OPTIMUM SUBMARINE ROUTING II
COMPUTATIONAL ROUTINES

GEORGE D. SCHMIEG

down on computer time, a search was made up over only the two variables v,w, since u, the heading angle, is of relatively small significance in the computation of the Hamiltonian. The search consisted of taking all possible combinations of v,w where v ranged over the set of values 3,6, 9,12,15 and w ranged over the values 200,400,600,800,1000 and comparing the Hamiltonian for each set with the one generated in the numerical routine. It is possible for the search with this size grid to fail to detect a set of controls that yield a smaller value for the Hamiltonian. However, the time factor is critical and this grid size was felt to furnish a satisfactory compromise. A search relying on gross computation can easily become completely unrealistic in terms of the computer time required.

The Hamiltonian

$$(3.9) \qquad H = \vec{\Lambda} \cdot \vec{V}$$

has a constant value on an extremal, with allowance being made for round off errors, whenever t does not occur explicitly in H.

At this point, we might mention that there are two types of variations of the control variables in the classical literature, called weak variations and strong variations. [4] Weak variations are variations in which the $|\delta u^i|$ are "small" for each time step; strong variations are variations in which $\int_0^T |\delta u^i| \, dt$ is "small". That is, in weak variations only values of control near those used are compared but if strong variations are considered, then the new control function may not be "near" the one used.

All methods of determining the routes are methods of variations which deform a given path. A path which furnishes a relative minimum, if we allow only weak variations, may not furnish such a minimum if strong variations are allowed, as will be seen in section 6.

5. The Numerical Routine.

The routine for determining the route is given in this section.

Heuristic Discussion:

Let us guess a set of values for the parameters h_1, h_2. We will then use this set of values to determine the control variables for each time t by the minimum principle to determine a route. The terminal point thus generated will, in general, differ from the desired one. By changing the values of h_1, h_2 appropriately, this terminal point will be forced toward the desired point x_T, y_T.

Mathematical Derivation:

First we see from (4.3) that

$$\delta\vec{\Lambda} = \vec{\Lambda}_1 dh_1 + \vec{\Lambda}_2 dh_2. \tag{5.1}$$

Note that $\nu = 1$, and this in turn implies $\delta\nu = 0$. These facts will be used in the following equations.

Next, from the first of the Euler equations (4.4) by taking the total differential we find that

$$\begin{aligned} &- v \sin u\,\delta\lambda + v \cos u\,\delta\mu + (-\lambda v \cos u - \mu v \sin u + f_{uu})\,\delta u \\ &+(-\lambda \sin u - \mu \cos u + f_{uv})\delta v + f_{uw}\delta w = 0, \end{aligned} \tag{5.2}$$

if we assume that we can change λ, μ, u, v, w, at fixed x, y, z, t. But since

$$H_{uu} = -\lambda v \cos u - \mu v \sin u + f_{uu}$$

$$H_{uv} = -\lambda \sin u - \mu \cos u + f_{uv}$$

$$H_{uw} = f_{uw}$$

(5.2) becomes

5-5-3

OPTIMUM SUBMARINE ROUTING II

COMPUTATIONAL ROUTINES

by

George D. Schmieg
Lieutenant Junior Grade, United States Naval Reserve
B.A., State University of South Dakota, 1963

Submitted in partial fulfillment

for the degree of

MASTER OF SCIENCE

WITH MAJOR IN MATHEMATICS

from the

UNITED STATES NAVAL POSTGRADUATE SCHOOL
August 1966

ABSTRACT

In this paper the details of computing an optimum route for a submarine are studied.

Typical functions representing the listening devices were used. It was found that in some cases several extremals existed and it was necessary to set up tests for the Legendre and Weierstrass conditions. The problem is further complicated by the fact that the optimum control-variables may lie on the boundary of the region of allowed values and further routines must be adjoined for this. Further, corners may occur and in particular the control may move discontinuously from a boundary point to an interior point or vice versa. The routines were made up to effect a compromise between the need for accuracy and reasonable computational time.

TABLE OF CONTENTS

Introduction

The purpose of this paper was to develop a numerical routine for solving the submarine routing problem; the problem is for the submarine to choose a course that minimizes the probability that it will be detected. Several difficulties arise in the numerical solution and subroutines were included to take care of these.

In type, the problem is a problem of Bolza in calculus of variations. It is complicated by the fact that there may be several routes each of which appear to be the solution. They all begin and end at the desired points and they all satisfy the Euler equations. It is only after routines are incorporated to check other conditions, the conditions of Legendre and Weierstrass, that it becomes clear whether a particular solution is the desired solution. When these conditions are not satisfied this fact must be determined and a routine made up to determine the controls to satisfy it.

In general, the route is generated as an extremal, a solution to the Euler equations, though the basic principle is that the control must minimize the Hamiltonian. The problem is complicated by the fact that the control which furnishes the minimum may lie on the boundary to the region, as when the submarine is at maximum depth. Several subroutines must be adjoined to treat the case when the control lies on one of the bounding faces or edges.

The existence of a corner introduces further complications; at a corner the control is a discontinuous function of time. It is necessary to make up a search routine for other values of the control which may decrease the Hamiltonian, and it is necessary to compromise between the demands of computing time and accuracy in this routine.

I am indebted to my advisor Professor F. D. Faulkner for the encouragement and guidance given me. This is a continuation of the study [1] which he began. The principal purpose of this study was to analyze the numerous difficulties envisioned and to make up numerical routines to resolve these and others that developed, so that a solution could be generated automatically on the computer. I also express my thanks to Professor W.E. Bleick, whose guidelines were followed in programming the numerical routine associated with this problem, and to Mrs. Sally B. Kline for help when programming difficulties were encountered.

1. Statement of the Problem.

The basic problem studied here is the following. A submarine located at point x_0, y_0 is to make a voyage to point x_T, y_T in a specified time T. Throughout this voyage the submarine is subjected to enemy detection devices whose capabilities are assumed known statistically. If the submarine has previously gone undetected, let the probability of detection in a time interval Δt be approximately $f(x,y,u,v,w,t)\,\Delta t$, and hence the probability of being detected along the route satisfies the equation

$$dp = (1 - p)\, f(x,y,u,v,w,t)\, dt. \tag{1.1}$$

The function f is the best estimate of the enemy's detection capabilities based on information we have about his listening devices, tests we have run on our submarines using comparable devices, the distance involved in the trip, and other information available to us.

Equation (1.1) can be simplified by letting

$$z = -\ln(1 - p) \tag{1.2}$$

which gives

$$\dot{z} = f. \tag{1.3}$$

Since we are primarily interested in long routes, and the time to change depth, speed, and heading angle is assumed small compared to total time, it will be ignored.

2. Equations.

Because routes of approximately 2500 miles are of primary interest, the flat earth assumption will be used. The equations governing the system may then be written as

$$\begin{aligned} \dot{x} &= v \cos u \\ \dot{y} &= v \sin u \qquad (2.1) \\ \dot{z} &= f(x,y,u,v,w,t). \end{aligned}$$

limited range. The constant c_2 reflects the possibility of detection by some passing ship, for example.

The thermocline is defined as the depth at which the temperature gradient (rate of decrease of temperature with increasing depth) is a maximum. In equations (2.3), w_0 is the depth of the thermocline, s_1 is the distance from the terminal point (x_T,y_T) of the route to the main concentration of enemy listening devices which is represented by the point (x_2,y_2), and θ is the angle that the perpendicular to the enemy shore line makes with the x-axis. The constants a_1,b_1,c_1,d_1, and w_0 are chosen so as to simulate, by the function f, conditions as they exist in any given situation.

The two functions f_1 and f_2 are intended to be typical of the functions which do describe the enemy's defenses. The function f_1, representing the passive enemy defense, tends to decrease as the distance from the enemy shoreline increases. This mathematical model of the enemy's passive defenses also tends to be insensitive in the region of the thermocline. The function f_2, which represents the enemy active defense, was constructed so as to emphasize the enemy's surface search. For this reason, this function decreases as the depth increases.

If submarine routing as described in this paper were to be made a part of naval operations, generating the function f would be a problem for intelligence and engineers. Such things as the sea state and the nature of the ocean floor in the region where the submarine is operating would then have to be included in the function f. These functions would also vary with time, reflecting changing sea state, etc.

The problem now becomes that of determining the control variables as functions of time to effect the desired optimization. That is, we want to choose the heading, the speed, and the depth so as to go from the initial point (x_0,y_0) to the terminal point (x_T,y_T) with the minimum probability of detection.

By (1.1) the probability of being detected along the route is

$$p(T) = 1 - \exp\left[- z(T)\right] \tag{2.5}$$

and hence this is the quantity we wish to minimize.

3. Adjoint Equations.

Let us consider any route and a neighboring route. The neighboring route will be generated by replacing u,v,w on the original route by $u + \delta u$, $v + \delta v$, $w + \delta w$ respectively. This generates first-order changes in x,y,z satisfying the differential equations

$$\begin{aligned} \delta\dot{x} &= - v \sin u\, \delta u + \cos u\, \delta v \\ \delta\dot{y} &= v \cos u\, \delta u + \sin u\, \delta v \\ \delta\dot{z} &= f_u\, \delta u + f_v\, \delta v + f_w\, \delta w + f_x\, \delta x + f_y\, \delta y. \end{aligned} \tag{3.1}$$

The notation $f_u \equiv \frac{\partial f}{\partial u}$, etc. as used above will be employed throughout the remainder of the paper.

Let us now introduce three Lagrange multipliers λ, μ, ν which are some unspecified function of time t. Multiply equations (3.1) by λ, μ, ν in that order, add the three equations, and integrate the result over the interval (0,T). These operations yield

$$\int_0^T \Big[\lambda(\delta\dot{x} + v \sin u\, \delta u - \cos u\, \delta v + \mu(\delta\dot{y} - v \cos u\, \delta u - \sin u\, \delta v) + \nu(\delta\dot{z} - f_x \delta x - f_y \delta y - f_u\, \delta u - f_v \delta v - f_w \delta w)\Big]\, dt. \tag{3.2}$$

Separating the terms containing the variations of the state variables from those containing the variations of the control variables, we get

$$\int_0^T \left[\lambda\,\delta\dot{x} + \mu\,\delta\dot{y} + \nu(\delta\dot{z} - f_x\delta x - f_y\delta y)\right] dt$$

$$= \int_0^T \Big(\left[-\lambda v \sin u + \mu v \cos u + \nu f_u\right]\delta u + \left[\lambda \cos u + \mu \sin u + \nu f_v\right]\delta v + \nu f_w\,\delta w\Big)\, dt. \tag{3.3}$$

Integrating by parts in (3.3) just the terms on the left involving derivatives of state variables, we get

$$\left[\lambda\,\delta x + \mu\,\delta y + \nu\,\delta z\right]_0^T - \int_0^T \left[\delta x(\dot{\lambda} + \nu f_x) + \delta y(\dot{\mu} + \nu f_y) + \delta z\,\dot{\nu}\right] dt \tag{3.4}$$

which combined with the right-hand equations of (3.3) yields

$$\left[\lambda\,\delta x + \mu\,\delta y + \nu\,\delta z\right]_0^T - \int_0^T \Big[\delta x(\dot{\lambda} + \nu f_x) + \delta y(\dot{\mu} + \nu f_y) + \delta z\,\dot{\nu}\; dt = \int_0^T \Big[(-\lambda v \sin u + \mu v \cos u + \nu f_u)\,\delta u + (\lambda \cos u + \mu \sin u + \nu f_v)\,\delta v + \nu f_w\,\delta w\Big] dt. \tag{3.5}$$

Now, choose λ, μ, ν in such a manner that they satisfy the differential equations

$$\begin{aligned} \dot{\lambda} &= -\nu f_x \\ \dot{\mu} &= -\nu f_y \\ \dot{\nu} &= 0. \end{aligned} \tag{3.6}$$

Equations (3.6) are called the adjoint equations of the variational equations (3.1). With this choice of λ, μ, ν equations (3.1) reduce to

$$\Big[\lambda\,\delta x + \mu\,\delta y + \nu\,\delta z\Big]_0^T = \int_0^T \Big[\lambda v \sin u - \mu v \cos u$$

$$(3.7) \qquad -\nu f_u\Big]\,\delta u\,dt + \int_0^T \Big[\lambda \cos u + \mu \sin u + \nu f_v\Big]\,\delta v\,dt +$$

$$\int_0^T \nu f_w\,\delta w\,dt.$$

Note that this formula gives us a relation between the terminal values of x,y,z and an integral made up of the variations of the control variables δu, δv, δw. Note also the important fact that the values of δx, δy, δz interior to the interval (0,T) are not needed.

For convenience in the sections to follow, the vector notation

$$\vec{\Lambda} = \begin{pmatrix} \lambda \\ \mu \\ \nu \end{pmatrix}$$

$$(3.8) \qquad \vec{V} = \begin{pmatrix} v \cos u \\ v \sin u \\ f \end{pmatrix}$$

will be used. The scalar product of these two vectors

$$(3.9) \qquad H = \vec{\Lambda} \cdot \vec{V}$$

is called the Hamiltonian.

Also for future use, let us define the following three particular solutions to the adjoint equations

$$\vec{\Lambda}_1 = \begin{pmatrix} \lambda_1 \\ \mu_1 \\ \nu_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$(3.10) \qquad \vec{\Lambda}_2 = \begin{pmatrix} \lambda_2 \\ \mu_2 \\ \nu_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\vec{\Lambda}_3 = \begin{pmatrix} \lambda_3 \\ \mu_3 \\ \nu_3 \end{pmatrix} = \begin{pmatrix} -\int_0^T f_x \, dt \\ -\int_0^T f_y \, dt \\ 1 \end{pmatrix}$$

4. Conditions for a Minimum.

In this section, the necessary condition for a minimum will be given.

If a path is to provide the desired minimum, it must first be admissible.

Admissibility. One requirement for a route to be admissible is that it begin and end at the desired points, i.e., $x(0) = 0$, $y(0) = 0$ and $x(T) = x_T$, $y(T) = y_T$ for some solution to the differential equations

$$\begin{aligned} \dot{x} &= v \cos u \\ \dot{y} &= v \sin u \qquad (2.1) \\ \dot{z} &= f. \end{aligned}$$

A further condition for admissibility is that the depth and speed satisfy the inequalities

$$\begin{aligned} &0 \leq w \leq w_{\max} \qquad (4.1) \\ &0 \leq v \leq v_{\max} \end{aligned}$$

where $w_{\max}$ and $v_{\max}$ depend upon the specific class of submarine under consideration.

A set of control variables which are piecewise continuous and satisfy (4.1) are called allowable. Note that allowability is a local constraint, in terms of time, on the set of control variables. A path which satisfies the above constraints, (2.1), (4.1), is admissible.

Our problem is to find the one route, among all admissible routes, such that z(T) is a minimum. Minimizing z(T) in turn minimizes p(T) and this is our objective.

For a path to furnish a minimum, it must be admissible and also satisfy the following conditions:

1. Euler Equations.
2. Legendre Condition.
3. Weierstrass Condition.
4. Envelope Condition.

The envelope condition was not investigated and will not be treated in this paper. The order of the conditions as given above is used since this is the usual order in which they will be checked in a problem.

A point that should be kept in mind throughout is that on a path which furnishes the desired minimum, the Hamiltonian must be minimized at each value of t. This is the Weierstrass-Pontryagin maximum principle with a change of sign. To satisfy this one criterion, the Euler equations, the Legendre condition, and the Weierstrass condition must all be satisfied.

The Euler equations, a condition on the first derivatives, require that the Hamiltonian have a stationary value.

In the Legendre condition the second derivatives are checked for a minimum. [2] The Euler equations and the Legendre conditions are both local conditions on the control variables u,v,w.

Finally, in the Weierstrass condition we compare the Hamiltonian for the values of u,v,w used with the Hamiltonian for all allowable values of u,v,w, for all values of t.

Euler Equations. For a minimum, the control variables must be chosen so they minimize

$$(3.9) \qquad H = \vec{\Lambda} \cdot \vec{V}$$

as compared with all allowable controls, for all t, $0 < t < T$, for some solution $\vec{\Lambda}$ to the adjoint system of equations. Let us consider our solution $\vec{\Lambda}$ to the adjoint equations in the form

$$(4.2) \qquad \vec{\Lambda} = h_1 \vec{\Lambda}_1 + h_2 \vec{\Lambda}_2 + h_3 \vec{\Lambda}_3$$

where h_1, h_2, h_3 are arbitrary constants and $\vec{\Lambda}_1$, $\vec{\Lambda}_2$, $\vec{\Lambda}_3$ are defined in equations (3.10).

We are now faced with the problem of having three constants in our solution $\vec{\Lambda}$ to the adjoint equations. But we may choose one relation among the constants h_1, h_2, h_3. We chose h_3 to be 1, and then

$$(4.3) \qquad \vec{\Lambda} = h_1 \vec{\Lambda}_1 + h_2 \vec{\Lambda}_2 + \vec{\Lambda}_3.$$

This then leaves us with the problem of finding the other two constants so that x(T), y(T) will assume the desired terminal values x_T, y_T.

If the values of the control variables lie inside the domain of allowed values, then the Euler equations

$$\begin{aligned} &\frac{\partial H}{\partial u} = H_u = -\lambda v \sin u + \mu v \cos u + \nu f_u = 0 \\ (4.4) \qquad &\frac{\partial H}{\partial v} = H_v = \lambda \cos u + \mu \sin u + \nu f_v = 0 \\ &\frac{\partial H}{\partial w} = H_w = \nu f_w = 0 \end{aligned}$$

are the first necessary conditions for the desired minimum. The Euler equations when combined with the adjoint equations (3.6) are referred to as the Euler-Lagrange equations. Solutions to the equations of motion which also satisfy the Euler-Lagrange equations are called <u>extremals.</u>

The Euler equations,however,do not insure that we effect the desired minimum for H; they are only necessary conditions. It is then necessary to investigate additional conditions, the next being the Legendre condition.

Legendre Condition. This is an investigation of the second-degree terms of the Taylor expansion of the Hamiltonian. If we can expand H in a series in du, dv, dw, valid in some neighborhood of (0,0,0), the first necessary condition for H to be a minimum is that $H_u = H_v = H_w = 0$, as stated in equations (4.4). The second necessary condition is that the quadratic form

$$(4.5) \qquad (du\ dv\ dw) \begin{pmatrix} H_{uu} & H_{uv} & H_{uw} \\ H_{vu} & H_{vv} & H_{vw} \\ H_{wu} & H_{wv} & H_{ww} \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix}$$

be positive semi-definite at least; we hope it will be positive definite. This condition, which is known as the Legendre condition, [3] may be expressed

$$H_{uu} > 0$$

$$\begin{vmatrix} H_{uu} & H_{uv} \\ H_{vu} & H_{vv} \end{vmatrix} > 0$$

$$(4.6) \qquad \begin{vmatrix} H_{uu} & H_{uv} & H_{uw} \\ H_{vu} & H_{vv} & H_{vw} \\ H_{wu} & H_{wv} & H_{ww} \end{vmatrix} > 0$$

for the case in which the control variables u,v,w all lie interior to the domain of allowed values.

If one or more of the control variables are on the boundary, the conditions are altered accordingly. Consider the case in which $w = w_{max}$ for some part of the route. The conditions then become

$$H_w < 0$$

$$(4.7) \qquad \begin{vmatrix} H_{uu} & H_{uv} \\ H_{vu} & H_{vv} \end{vmatrix} > 0$$

whenever $w = w_{max}$. If $w = w_{max}$ and $H_w = 0$, then we must check to ensure that $H_{ww} > 0$. Of course if $H_w > 0$ the minimum is not on the boundary at that point.

If $v = v_{max}$, conditions equivalent to those above will be used. When both $w = w_{max}$ and $v = v_{max}$, the conditions read

$$(4.8) \qquad \begin{aligned} &H_w < 0 \\ &H_v < 0 \\ &H_{uu} > 0, \end{aligned}$$

and if $H_w = 0$ on the boundary then we must check $H_{ww} > 0$, and similarly for H_v and H_{vv}.

If the Euler equations are satisfied and if equations (4.6) are satisfied at some interior point of u,v,w space, then these values yield a local minimum; H is smaller at that point than at any other point u,v,w in some neighborhood. However, the point may not furnish the minimum value for H; it is necessary in theory to compare the value with the value for all allowable values of u,v,w. This condition, that the value chosen minimize H, is called the Weierstrass condition.

A comparison must be made between the Hamiltonian for the u,v,w generated, H(u,v,w), and for all other allowable combinations of control variables, say, u^i, v^i, w^i. If $H(u,v,w) > H(u^i, v^i, w^i)$, then the control variables u,v,w must be replaced by the new set u^i, v^i, w^i, which yield the smaller value for the Hamiltonian.

Unfortunately there is no easy way to check this condition. To cut down on computer time, a search was made up over only the two variables v,w, since u, the heading angle, is of relatively small significance in the computation of the Hamiltonian. The search consisted of taking all possible combinations of v,w where v ranged over the set of values 3,6, 9,12,15 and w ranged over the values 200,400,600,800,1000 and comparing the Hamiltonian for each set with the one generated in the numerical routine. It is possible for the search with this size grid to fail to detect a set of controls that yield a smaller value for the Hamiltonian. However, the time factor is critical and this grid size was felt to furnish a satisfactory compromise. A search relying on gross computation can easily become completely unrealistic in terms of the computer time required.

The Hamiltonian

$$(3.9) \qquad H = \vec{\Lambda} \cdot \vec{V}$$

has a constant value on an extremal, with allowance being made for round off errors, whenever t does not occur explicitly in H.

At this point, we might mention that there are two types of variations of the control variables in the classical literature, called weak variations and strong variations. [4] Weak variations are variations in which the $|\delta u^i|$ are "small" for each time step; strong variations are variations in which $\int_0^T |\delta u^i|\,dt$ is "small". That is, in weak variations only values of control near those used are compared but if strong variations are considered, then the new control function may not be "near" the one used.

All methods of determining the routes are methods of variations which deform a given path. A path which furnishes a relative minimum, if we allow only weak variations, may not furnish such a minimum if strong variations are allowed, as will be seen in section 6.

5. The Numerical Routine.

The routine for determining the route is given in this section.

Heuristic Discussion:

Let us guess a set of values for the parameters h_1, h_2. We will then use this set of values to determine the control variables for each time t by the minimum principle to determine a route. The terminal point thus generated will, in general, differ from the desired one. By changing the values of h_1, h_2 appropriately, this terminal point will be forced toward the desired point x_T, y_T.

Mathematical Derivation:

First we see from (4.3) that

$$\delta\vec{\Lambda} = \vec{\Lambda}_1 dh_1 + \vec{\Lambda}_2 dh_2. \tag{5.1}$$

Note that $\nu = 1$, and this in turn implies $\delta\nu = 0$. These facts will be used in the following equations.

Next, from the first of the Euler equations (4.4) by taking the total differential we find that

$$\begin{aligned} &- v \sin u\,\delta\lambda + v \cos u\,\delta\mu + (-\lambda v \cos u - \mu v \sin u + f_{uu})\,\delta u \\ &+(-\lambda \sin u - \mu \cos u + f_{uv})\delta v + f_{uw}\delta w = 0, \end{aligned} \tag{5.2}$$

if we assume that we can change λ, μ, u, v, w, at fixed x, y, z, t. But since

$$H_{uu} = -\lambda v \cos u - \mu v \sin u + f_{uu}$$

$$H_{uv} = -\lambda \sin u - \mu \cos u + f_{uv}$$

$$H_{uw} = f_{uw}$$

(5.2) becomes

$$(5.3) \qquad -v \sin u\, \delta\lambda + v \cos u\, \delta\mu + H_{uu}\, \delta u + H_{uv}\, \delta v + H_{uw}\, \delta w = 0.$$

Similarly

$$(5.4) \qquad \begin{aligned} &\cos u\, \delta\lambda + \sin u\, \delta\mu + (-\lambda \sin u + \mu \cos u + f_{vu})\, \delta u \\ &+ f_{vv}\, \delta v + f_{vw}\, \delta w = 0 \end{aligned}$$

and since

$$H_{vu} = -\lambda \sin u + \mu \cos u + f_{vu}$$

$$H_{vv} = f_{vv}$$

$$H_{vw} = f_{vw}$$

(5.4) becomes

$$(5.5) \qquad \cos u\, \delta\lambda + \sin u\, \delta\mu + H_{vu}\, \delta u + H_{vv}\, \delta v + H_{vw}\, \delta w = 0.$$

The third Euler equation gives us

$$(5.6) \qquad f_{wu}\, \delta u + f_{wv}\, \delta v + f_{ww}\, \delta w = 0.$$

But since

$$f_{wu} = H_{wu}$$

$$f_{wv} = H_{wv}$$

$$f_{ww} = H_{ww}$$

(5.6) becomes

$$(5.7) \qquad H_{wu}\, \delta u + H_{wv}\, \delta v + H_{ww}\, \delta w = 0.$$

Now consider the equations (5.3), (5.5), (5.7) as three equations in the three unknowns δu, δv, δw.

$$(5.8) \qquad \begin{aligned} H_{uu}\, \delta u + H_{uv}\, \delta v + H_{uw}\, \delta w &= v \sin u\, \delta\lambda - v \cos u\, \delta\mu \\ H_{vu}\, \delta u + H_{vv}\, \delta v + H_{vw}\, \delta w &= -\cos u\, \delta\lambda - \sin u\, \delta\mu \\ H_{wu}\, \delta u + H_{wv}\, \delta v + H_{ww}\, \delta w &= 0. \end{aligned}$$

If the determinant of coefficients of δu, δv, δw in (5.8) does not vanish, we can solve for δu, δv, δw by using Cramer's Rule as follows:

Denote the determinant of coefficients by D, i.e.,

$$(5.9)\qquad D = \begin{vmatrix} H_{uu} & H_{uv} & H_{uw} \\ H_{vu} & H_{vv} & H_{vw} \\ H_{wu} & H_{wv} & H_{ww} \end{vmatrix} .$$

Then

$$\delta u = \frac{\begin{vmatrix} (v \sin u\, \delta\lambda - v \cos u\, \delta\mu) & H_{uv} & H_{uw} \\ (- \cos u\, \delta\lambda - \sin u\, \delta\mu) & H_{vv} & H_{vw} \\ 0 & H_{wv} & H_{ww} \end{vmatrix}}{D}$$

$$(5.10)$$

$$\delta v = \frac{\begin{vmatrix} H_{uu} & (v \sin u\, \delta\lambda - v \cos u\, \delta\mu) & H_{uw} \\ H_{vu} & (- \cos u\, \delta\lambda - \sin u\, \delta\mu) & H_{vw} \\ H_{wu} & 0 & H_{ww} \end{vmatrix}}{D}$$

Notice that δw could be found in the same manner, but is not needed since it is not used in the numerical routine. Next, since from (5.1) $\delta\lambda = dh_1$ and $\delta\mu = dh_2$

$$\delta u = \frac{-(v \sin u\, dh_1 - v \cos u\, dh_2)\,(H_{vv} H_{ww} - H_{wv} H_{vw})}{D}$$

$$(5.11)\qquad \frac{-\,(\cos u\, dh_1 + \sin u\, dh_2)\,(H_{uv} H_{ww} - H_{wv} H_{uw})}{D}$$

$$\delta v = \frac{(v \sin u \; dh_1 - v \cos u \; dh_2)\,(H_{vu}\, H_{ww} - H_{wu}\, H_{vw})}{D} + \frac{(\cos u \; dh_1 + \sin u \; dh_2)\,(H_{uu}\, H_{ww} - H_{wu}\, H_{uw})}{D} .$$

To simplify the form of the equation let us set

$$\begin{aligned}
S^{11} &= \Big[(H_{vv}\, H_{ww} - H_{wv}\, H_{vw})\,(-\, v \sin u) + (H_{uv}\, H_{ww} - H_{wv}\, H_{uw})\,(-\cos u)\Big] \,/\, D \\
S^{12} &= \Big[(H_{vv}\, H_{ww} - H_{wv}\, H_{vw})\,(v \cos u) + (H_{uv}\, H_{ww} - H_{wv}\, H_{uw})\,(-\sin u)\Big] \,/\, D \\
S^{21} &= \Big[(H_{vu}\, H_{ww} - H_{wu}\, H_{vw})\,(v \sin u) + (H_{uu}\, H_{ww} - H_{wu}\, H_{uw})\,(\cos u)\Big] \,/\, D \\
S^{22} &= \Big[(H_{vu}\, H_{ww} - H_{wu}\, H_{vw})\,(-\, v \cos u) + (H_{uu}\, H_{ww} - H_{wu}\, H_{uw})\,(\sin u)\Big] \,/\, D;
\end{aligned} \tag{5.12}$$

then the above equations may be rewritten

$$\begin{aligned}
\delta u &= S^{11}\, dh_1 + S^{12}\, dh_2 \\
\delta v &= S^{21}\, dh_1 + S^{22}\, dh_2
\end{aligned} \tag{5.13}$$

We now have the machinery to derive $\delta x(T)$, $\delta y(T)$. In deriving $\delta x(T)$, we use equations (3.5) and a particular choice for $\vec{\Lambda}$, namely the $\vec{\Lambda}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ as defined in equations (3.10). This choice for $\vec{\Lambda}$ in equation (3.7) yields

$$\Big[\delta x\Big]_0^T = \int_0^T \cos u \;\delta v \; dt - \int_0^T v \sin u \;\delta u \; dt$$

which becomes, after noting that $\delta x(0) = 0$ and substituting δu, δv from equations (5.13),

$$\text{(5.14)} \qquad \delta x(T) = \int_0^T \Big[\cos u\,(S^{21}\,dh_1 + S^{22}\,dh_2) - v \sin u\,(S^{11}\,dh_1 + S^{12}\,dh_2)\Big]\,dt.$$

Similarly, using $\vec{\Lambda}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, we find

$$\text{(5.15)} \qquad \delta y(T) = \int_0^T \Big[\sin u\,(S^{21}\,dh_1 + S^{22}\,dh_2) + v \cos u\,(S^{11}\,dh_1 + S^{12}\,dh_2)\Big]\,dt.$$

Equations (5.14) and (5.15) can be rewritten as

$$\text{(5.16)} \qquad \begin{aligned} \delta x(T) &= dh_1 \int_0^T \Big[\cos u\, S^{21} - v \sin u\, S^{11}\Big]\,dt + dh_2 \int_0^T \Big[\cos u\, S^{22} - v \sin u\, S^{12}\Big]\,dt \\ \delta y(T) &= dh_1 \int_0^T \Big[\sin u\, S^{21} + v \cos u\, S^{11}\Big]\,dt + dh_2 \int_0^T \Big[\sin u\, S^{22} + v \cos u\, S^{12}\Big]\,dt. \end{aligned}$$

If we set

$$A_{11} = \int_0^T (\cos u\, S^{21} - v \sin u\, S^{11})\,dt$$

$$A_{12} = \int_0^T (\cos u\, S^{22} - v \sin u\, S^{12})\,dt$$

$$A_{21} = \int_0^T (\sin u\, S^{21} + v \cos u\, S^{11})\, dt$$

$$A_{22} = \int_0^T (\sin u\, S^{22} + v \cos u\, S^{12})\, dt,$$

equations (5.16) become

$$\begin{aligned} \delta x(T) &= A_{11}\, dh_1 + A_{12}\, dh_2 \\ \delta y(T) &= A_{21}\, dh_1 + A_{22}\, dh_2 . \end{aligned} \tag{5.17}$$

Equations (5.17) give us the mechanism for a Newton - Raphson iteration scheme for correcting h_1, h_2, if we make the following substitutions

$$\begin{aligned} \delta x(T) &= x_T - x(T) \\ \delta y(T) &= y_T - y(T) . \end{aligned} \tag{5.18}$$

Note that in equations (5.18) $\delta x(T)$, $\delta y(T)$ were equated to the desired terminal values minus those which were attained. Using equations (5.18), equations (5.17) become

$$\begin{aligned} x_T - x(T) &= A_{11}\, dh_1 + A_{12}\, dh_2 \\ y_T - y(T) &= A_{21}\, dh_1 + A_{22}\, dh_2 \end{aligned} \tag{5.19}$$

from which we are able to generate corrections for the constants h_1, h_2.

Note above that in equations (5.19) the coefficients A_{11}, A_{12}, A_{21}, A_{22} are integrals with respect to time. In the numerical routine, this integration is accomplished by a Runge-Kutta numerical integration scheme, contained within which is a Newton-Raphson iteration to generate the necessary changes in u,v,w to accomplish the integration.

The mathematical basis for the Newton-Raphson iteration scheme to generate corrections for u, v, w is the following. From the equations

$$\begin{aligned} H_{uu}\, du + H_{uv}\, dv + H_{uw}\, dw &= - H_u \\ H_{vu}\, du + H_{vv}\, dv + H_{vw}\, dw &= - H_v \end{aligned} \tag{5.20}$$

$$H_{wu}\, du + H_{wv}\, dv + H_{ww}\, dw = - H_w$$

we find

$$du = \frac{\begin{vmatrix} - H_u & H_{uv} & H_{uw} \\ - H_v & H_{vv} & H_{vw} \\ - H_w & H_{wv} & H_{ww} \end{vmatrix}}{D}$$

(5.21)
$$dv = \frac{\begin{vmatrix} H_{uu} & - H_u & H_{uw} \\ H_{vu} & - H_v & H_{vw} \\ H_{wu} & - H_w & H_{ww} \end{vmatrix}}{D}$$

$$dw = \frac{\begin{vmatrix} H_{uu} & H_{uv} & - H_u \\ H_{vu} & H_{vv} & - H_v \\ H_{wu} & H_{wv} & - H_w \end{vmatrix}}{D}$$

where

$$D = \begin{vmatrix} H_{uu} & H_{uv} & H_{uw} \\ H_{vu} & H_{vv} & H_{vw} \\ H_{wu} & H_{wv} & H_{ww} \end{vmatrix} .$$

The iteration scheme then has the form

(5.22)
$$\begin{aligned} u_n &= u_{n-1} + du_n \\ v_n &= v_{n-1} + dv_n \\ w_n &= w_{n-1} + dw_n \end{aligned}$$

where n is the index for the Newton-Raphson iteration; du, dv, dw are those found in equations (5.21).

6. Control Variables on the Boundary.

In section 4, it was noted that for a path to be admissible the control variables v, w had to satisfy the inequalities

$$(4.1) \qquad \begin{aligned} 0 \leq w \leq w_{max} \\ 0 \leq v \leq v_{max}. \end{aligned}$$

Let us consider the situation in which the depth w assumes the maximum depth w_{max} for part or all of the route and the other bounded control variable v remains within its prescribed bounds. We must amend the routine for determining the controls as follows. We take the maximum depth w_{max} of the type of submarine being considered and read this information into the program of the numerical routine which calculates the route. If the Newton-Raphson iterations as established in (5.22) yield a value for w which exceeds w_{max}, we set w equal to w_{max} and generate u and v, using the following iteration scheme. From our Newton-Raphson iteration equations,

$$(6.1) \qquad \begin{aligned} H_{uu}\, du + H_{uv}\, dv = - H_u \\ H_{vu}\, du + H_{vv}\, dv = - H_v, \end{aligned}$$

we get corrections to the controls,

$$(6.2) \qquad du = \frac{\begin{vmatrix} - H_u & H_{uv} \\ - H_v & H_{vv} \end{vmatrix}}{\begin{vmatrix} H_{uu} & H_{uv} \\ H_{vu} & H_{vv} \end{vmatrix}} \qquad dv = \frac{\begin{vmatrix} H_{uu} & - H_u \\ H_{vu} & - H_v \end{vmatrix}}{\begin{vmatrix} H_{uu} & H_{uv} \\ H_{vu} & H_{vv} \end{vmatrix}}$$

Then the iterations discussed in the previous section take the form

$$(6.3) \qquad \begin{aligned} u_n &= u_{n-1} + du_n \\ v_n &= v_{n-1} + dv_n \, . \end{aligned}$$

Thus we may modify our Newton-Raphson iteration scheme for the controls when the minimum value of H occurs for w on the boundary. The modified Newton-Raphson iterations generate successive corrections to u and v so as to produce an admissible path. The numerical routine is programmed so that it is possible for the depth to assume its maximum for some part of the route without remaining fixed at maximum depth after once assuming it. In section 7 we will see an optimum path which has the control on the boundary for part of the path; the submarine travels at maximum depth for a while, then comes up.

The subroutine for calculating u, v when $w = w_{max}$ is contained in Appendix I, part C. The above features can be seen by examining either the flow chart for subroutine BOUNDW or the subroutine itself which is given in part G of Appendix I.

Modifications similar to those made for w on the boundary would be made if $v = v_{max}$ for some part of the route. One additional change required in this case, which was not necessary when $w = w_{max}$, is that δv be set equal to zero in the numerical routine whenever $v = v_{max}$ and $H_v < 0$. $V = v_{max}$ and $H_v > 0$ imply that the velocity is decreasing or moving away from the bound and hence the numerical routine described in equations (5.22) for generating corrections to the control variables would be used whenever this is the case. It should be noted that setting δw equal to zero in the case of $w = w_{max}$ was not required since δw does not enter into the equations for the numerical routine in section 5.

When $v = v_{max}$ and $H_v < 0$, we get from equations

$$(6.4)\qquad \begin{aligned} H_{uu}du + H_{uw}dw &= - H_u \\ H_{wu}du + H_{ww}dw &= - H_w \end{aligned}$$

the corrections to the control variables u, w as

$$(6.5)\qquad du = \frac{\begin{vmatrix} - H_u & H_{uw} \\ - H_w & H_{ww} \end{vmatrix}}{\begin{vmatrix} H_{uu} & H_{uw} \\ H_{wu} & H_{ww} \end{vmatrix}} \qquad dw = \frac{\begin{vmatrix} H_{uu} & - H_u \\ H_{wu} & - H_w \end{vmatrix}}{\begin{vmatrix} H_{uu} & H_{uw} \\ H_{wu} & H_{ww} \end{vmatrix}}$$

Using the du, dw found in equations (6.5), we get equations for correcting u, w

$$(6.6)\qquad \begin{aligned} u_n &= u_{n-1} + du_n \\ w_n &= w_{n-1} + dw_n\ . \end{aligned}$$

If both $v = v_{max}$ and $w = w_{max}$ and it is also true that $H_v < 0$ and $H_w < 0$, we generate corrections to the control variable u by using the equation

$$(6.7)\qquad H_{uu}du = - H_u$$

from which

$$(6.8)\qquad du = \frac{- H_u}{H_{uu}}$$

and the iterations to correct u take the form

$$(6.9)\qquad u_n = u_{n-1} + du_n .$$

Both of these situations were encountered in generating Path IV which will be discussed in section 8. The results given there will indicate that the modifications needed when $v = v_{max}$ or $v = v_{max}$ and $w = w_{max}$ do not affect convergence of the Newton-Raphson iteration schemes in the numerical routine, because we will find that Path IV is admissible.

The flow chart for the subroutine BOUNDV, which is used when $v = v_{max}$, can be seen in part D of Appendix I. Part B of Appendix I contains the flow chart of subroutine VUW which takes care of the case where $v = v_{max}$ and $w = w_{max}$. The result of setting $\delta v = 0$ when $v = v_{max}$ and $H_v < 0$ can be seen by examining the flow chart of the numerical routine in part A of Appendix I. The deck listings of BOUNDV,VUW, and the numerical routine can all be found in part G of Appendix I.

7. Paths.

Our computations have established the existence of three extremals. The three paths all satisfy the Euler-Lagrange equations as outlined in section 4.

These three paths can best be compared by listing the contrasting points of the three. The areas in which the greatest difference appeared among the three routes are the probability of detection p(T), the depth w, and the constants (h_1,h_2) which yield admissibility.

The following is a list of the results for each path in the three areas just mentioned.

<u>Path I</u>

p(T)	$.15741 \times 10^{-3}$
w	thermocline $\simeq$ 200 feet
(h_1,h_2)	(-.00019, .00024)

<u>Path II</u>

p(T)	$.22725 \times 10^{-3}$

w	w_{max} up to 683 feet
(h_1,h_2)	(-.00038, .00048)

Path III

$p(T)$	$.23060 \times 10^{-3}$
w	525 to 725 feet
(h_1,h_2)	(-.00037, .00046)

Path I gives us an absolute minimum, i.e., a minimum under either weak or strong variations. Path II gives us a relative minimum if only weak variations are allowed. Path III is an extremal, but does not furnish a relative minimum under either weak or strong variations. We called Path III a worsimax path.

If any additional information concerning any one of the above paths is desired, copies of the three paths can be found in Appendix II. Given there is a printout of each path with the coordinates (x,y) denoting the submarine's location, the control variables u,v,w, and the probability of detection p(t) for each time step.

Note that in Path II $w = w_{max}$ for nearly all of the route. In calculating this path, the iteration scheme described in equations (6.1), (6.2), (6.3) was used. The fact that Path II converges to the desired terminal point (x_T,y_T) substantiates the results given in section 6.

Analysis of Paths. Checks on the conditions as outlined in section 4 point out the following facts. Path I satisfied the following conditions:

1. admissibility conditions,
2. Euler equations,
3. Legendre conditions, and
4. Weierstrass condition.

Path II satisfied

1. admissibility conditions,
2. Euler equations, and
3. Legendre conditions,

but not the Weierstrass condition.

Path III satisfied

1. admissibility conditions, and
2. Euler equations,

but not the Legendre conditions nor the Weierstrass condition.

Hence, by the criterion established in section 4, there is but one minimum path, that being Path I. Note that the check of the envelope condition was not included in this investigation.

The above results point out an important fact which is often ignored or overlooked; the generation of an extremal, by no means guarantees that you have the desired minimum. This emphasizes the need for a check on all of the conditions for a minimum at each time step of the numerical integration scheme for generating the path. If checking after each time step requires excessive computer time, the checks may be performed at some appropriate periodic intervals.

It can be noted at this time, that if we restrict ourselves to weak variations as defined in section 4, both Path I and Path II are extremals which yield relative minima. In contrast to this, analyzing the paths and considering strong variations yields the result noted above, namely, that Path I is really the only relative minimum among the three paths.

After Path II, which does not give a relative minimum, was generated, a search, as outlined in the discussion of the Weierstrass condition in section 4, was used to determine a new set of control variables u,v,w, which would satisfy the Weierstrass condition. The resulting paths turned out to converge to Path I. The subroutine search was used for this purpose and is contained in Appendix I.

A similar search over the grid outlined in section 4 was performed when the Legendre conditions were not met in the case of Path III. The set of control variables which gave the smallest value for the Hamiltonian in this search were then used to continue the numerical routine. Again the sequence of paths produced by the numerical integration converged to Path I.

A path was judged admissible if it came within one-fourth mile of the desired endpoint; it was felt that further accuracy was not worth the computing time.

To test the convergence of the numerical routine, on a few paths the routine was allowed to continue until no further improvement occurred. In each of the three paths above, duplication occurred with accuracy of at least one-tenth of a nautical mile. Duplication here means the ability of the numerical routine to repeat itself after once converging to the desired terminal point.

It has been noted that a spiral pattern of convergence about the desired terminal point is present in the computation of each path. It is not clear why this occurs but the following is offered as a possible explanation. In the computation, we take $H_u = H_v = H_w = 0$ and vary u,v, w,x,y, h_1,h_2, but we assume that x,y have the values they assume on the path. For the purposes of explanation, let us examine the equation $H_u = 0$, from which we get an equation of the form

$$H_{uh_1} dh_1 + H_{uh_2} dh_2 + H_{uu} \delta u + H_{uv} \delta v + H_{uw} \delta w +$$

$$H_{ux} \delta x + H_{uy} \delta y = 0$$

when we take its variations. The last two terms in this expression drop out in linear problems and can be shown to be negligible if T is small in any case. They introduce considerable complication and extra computation and hence were discarded. This omission may be the reason for the spiral pattern of convergence. It should be pointed out that the terms were omitted in only the correction routine. If the sequence of paths converges, there is no related error in the path to which they converge.

In generating Path II, a subroutine was used within which the heading was fixed and a search over the depth and the speed was conducted to determine which set of values for these two controls gave the smallest value for the Hamiltonian H. These values were then used in the numerical routine to insure a start in the proper direction. This method was put into use when it was found that poor initial choices for u,v,w caused the Newton-Raphson routine to diverge at the beginning of the route. This subroutine SEARCH can be seen in Appendix I, part G.

To insure a proper start in computing Path II, the subroutine WORSI was used. This subroutine is the same as subroutine BOUNDW described in section 6 with one exception, that being that $w = w_{max}$ is replaced by $w = 500$ or some other intermediate value for the depth. This subroutine then fixes w at 500 and computes u and v using the iteration scheme described in equations (6.3). The resulting set of values for u and v are then combined with $w = 500$ to make up the initial guesses for the numerical routine. This subroutine is also given in Appendix I, part G.

Few problems were encountered in the generation of Path I, but the introduction of conditions to cause the submarine to assume its maximum depth w_{max} or velocity v_{max} aggravated the situation and introduced difficulties to make the subroutines listed above necessary.

It should be realized that this problem contains some real difficulties, if approached blindly. With the proper background and forethought, most of the difficulties can be anticipated and handled, when encountered, by methods such as those described above.

8. Corner.

This section contains a discussion of a case in which the path generated contains a corner.

A corner appears when control variables u,v,w which minimize the Hamiltonian are discontinuous functions of t. For convenience, the notation

$$\vec{U} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

will be used to denote a set u,v,w of control variables. The conditions for a corner are, first, that there exists some point on the route, call it t_1, where two sets of control variables $\vec{U}^1$ and $\vec{U}^2$ both minimize the Hamiltonian, H, and second that one set of control variables, say, $\vec{U}^1$, gives a smaller value for H when $t < t_1$, whereas the other set of controls $\vec{U}^2$ yield a smaller H for $t > t_1$. These two conditions can be stated as, first

$$(8.1) \qquad H(\vec{U}^1) = H(\vec{U}^2) = \min_{\vec{U}} \qquad t = t_1$$

and second

$$(8.2) \qquad \begin{array}{ll} H(\vec{U}^1) < H(\vec{U}^2) & \text{for } t < t_1 \\ H(\vec{U}^1) > H(\vec{U}^2) & \text{for } t > t_1 \end{array}$$

for some neighborhood of t_1.

At a corner the numerical routine for the corrections should be amended by adding terms to handle the changes due to the variation of the corner time t_1.

The changes necessary are outlined in the earlier report "Optimum Submarine Routing" [1], section 6. Even without these correction terms, the numerical routine generated an admissible path which contained the corner, namely Path IV.

To construct a mathematical model in which a corner could be expected the functions described in section 2 were changed in the following manner. We replaced

$$\frac{1 + \frac{.1\, w}{w_0}}{1 + \frac{w^2}{w_0}}$$

in the equation for f_2 by

$$\frac{1}{1 + \frac{w^2}{w_1^2}}$$

with w_1 equal to five hundred feet. The constants in f_1, f_2, the equations representing the passive and active defenses respectively, were chosen in such a way that a corner could be anticipated.

The model used was one in which the passive defense was dominant for the beginning of the route, the two would become equal at approximately the middle of the route, and the active search would then dominate for the latter part of the route. These facts are apparent when one looks at Path IV in Appendix II.

When we analyze Path IV, we see that the submarine proceeds at approximately thermocline depth for the first part of the route and then changes to $w = w_{max}$ for the remainder of the route. It can also be noted that the speed, v, was considerably less than $v = v_{max}$ until the corner was encountered, at which time v became equal to v_{max}. Whenever the active search completely dominates, the submarine goes as deep and as fast as possible.

A check on the Hamiltonian, H, after each time step shows that H is constant, within the accuracy of the routine, for the time steps before we reach the corner, but is not constant as we proceed beyond the corner. It is not clear why H does not remain constant throughout the route, but a possible explanation is that the corner was effectively passed before it was found, i.e., the numerical routine failed to detect the corner when the conditions for a corner were in fact present. The failure to find the corner immediately is a result of the grid size used in the subroutine SEARCH to compare the Hamiltonian for controls $\vec{U}$ generated by our numerical routine with the controls $\vec{U}^i$ used in the search. As noted before, this grid size was decided upon when a finer grid was found to require excessive computer time. Considering that it takes a while for the routine to find the corner and it takes the routine a certain amount of time to settle down after the corner is found, the fact that admissibility was accomplished was thought sufficient to justify omission of the corner correction terms.

Notice that in Path IV $v = v_{max}$ with $w < w_{max}$ for a part of the route before the corner and then $v = v_{max}$ and $w = w_{max}$ for the portion of the route that comes after the corner. Path IV was generated using the iteration schemes described in equations (6.4), (6.5), (6.6) when $v = v_{max}$ and $w < w_{max}$ and by using (6.7), (6.8), (6.9) when $v = v_{max}$ and $w = w_{max}$. The admissibility of Path IV confirms the results stated in section 6 for v on the boundary and when both v and w are on the bound of their allowable values.

9. Observations.

This section contains observations which may be helpful to a person wishing to continue the study of the submarine routing problem.

In the Newton-Raphson iterations which occur it may be possible to improve both convergence and accuracy as follows. For example, in generating the control variables let us make up an error function

$$e_1 H_u^2 + e_2 H_v^2 + e_3 H_w^2 .$$

If each successive iteration does not diminish this error function, then we should diminish the preceeding corrections by a factor of say, two, or five. The reason is that the Newton-Raphson iteration moves the variables in the right direction but may overshoot if the linear terms are not dominant. The iteration would be terminated when the above error function was less than some preassigned value. The incorporation of such routines might well improve convergence, save computing time, and improve accuracy. The convergence criterion above is derived from the fact that satisfaction of the Euler equations implies H_u, H_v, H_w are all equal to zero. Similar conditions could be established for the other Newton-Raphson iterations for generating corrections to the parameters h_1, h_2. In the iterations to correct these the established criterion would be based upon admissibility of the path. By letting (x,y) represent the endpoint of the path generated and (x_T, y_T) be the desired terminal point, we could write the condition as

$$(x - x_T)^2 + (y - y_T)^2 < \epsilon .$$

10. Summary.

In this paper the submarine routing problem was studied. Functions were chosen that seemed to be typical of the functions representing the detection devices, both passive and active. Information that could be arrived at only through the use of empirical data such as the sea state, for example, was not made a part of these functions.

Using this mathematical model and determining the path from (x_0,y_0) to (x_T,y_T), in a fixed time T, which minimizes the probability of detection, p(T), resulted in the generation of three extremals. Examination of the paths using established criterions for a relative minimum lead to the following results: one path yielded the desired minimum, a second satisfied all conditions except the Weierstrass condition and the third path was just an extremal, satisfying neither the Legendre nor the Weierstrass conditions.

Situations were encountered in which the speed, v, or the depth, w, or both v and w were on the boundary of allowable controls. It was found that if the control on the bound was set equal to the boundary value and corrections generated for the remaining control variables, admissibility was accomplished just as when all controls were interior to their allowable ranges.

With a change in the original model for the active defense, conditions conducive to a corner were established. The numerical routine then generated a path which had a corner and was admissible. Admissibility here was accomplished without the use of corrections for the corner, and for this reason the corrections were not made a part of the numerical routine.

The numerical routine as described in section 5 was programmed in

Fortran 1960 for a CDC 1604 computer. Both a flowchart and the program for this routine can be found in Appendix I, the flow chart in part A and the deck listing for the program in part G.

BIBLIOGRAPHY

1. Faulkner, F.D., "Optimum Submarine Routing". U.S. Naval Postgraduate School Research Paper No. 63, January 1966.

2. Leitmann, George, Optimization Techniques. New York, 1962, Chap. 1.

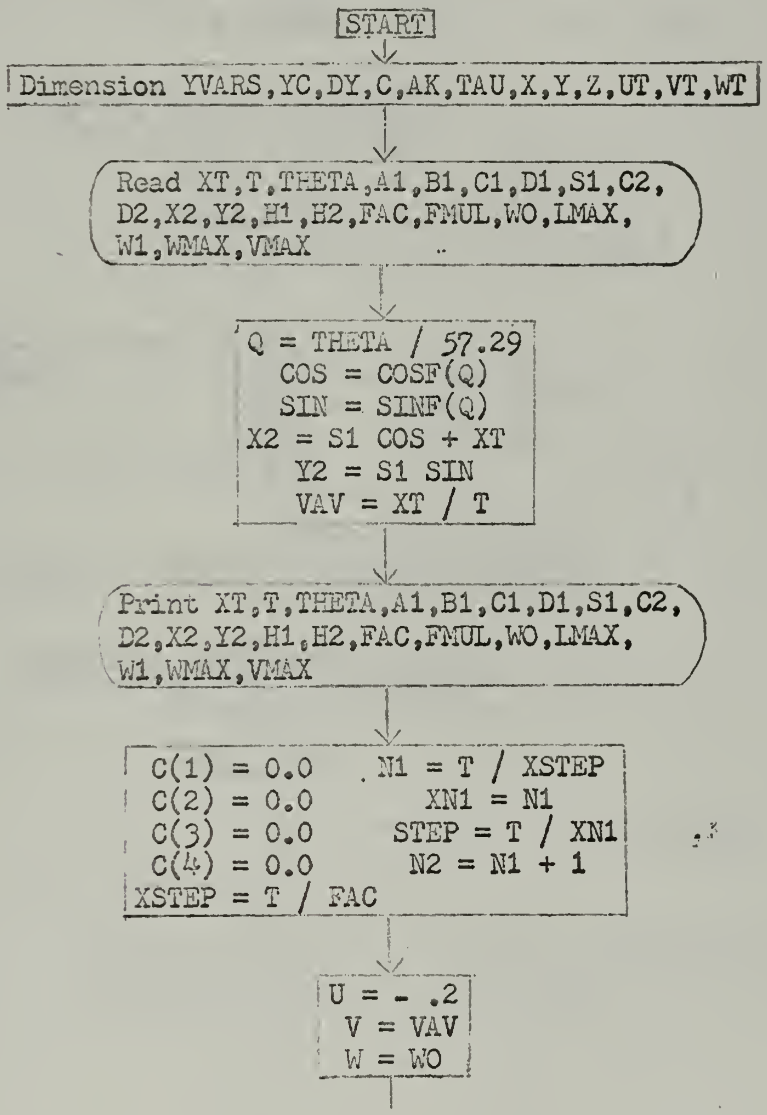
3. Bliss, G.A., Lectures on the Calculus of Variations, Chicago, 1946, p. 23.

4. Bolza, Oskar, Lectures on the Calculus of Variations, Chicago, 1904, Reprint Chelsea Publishing Co., New York, Chapter 3.

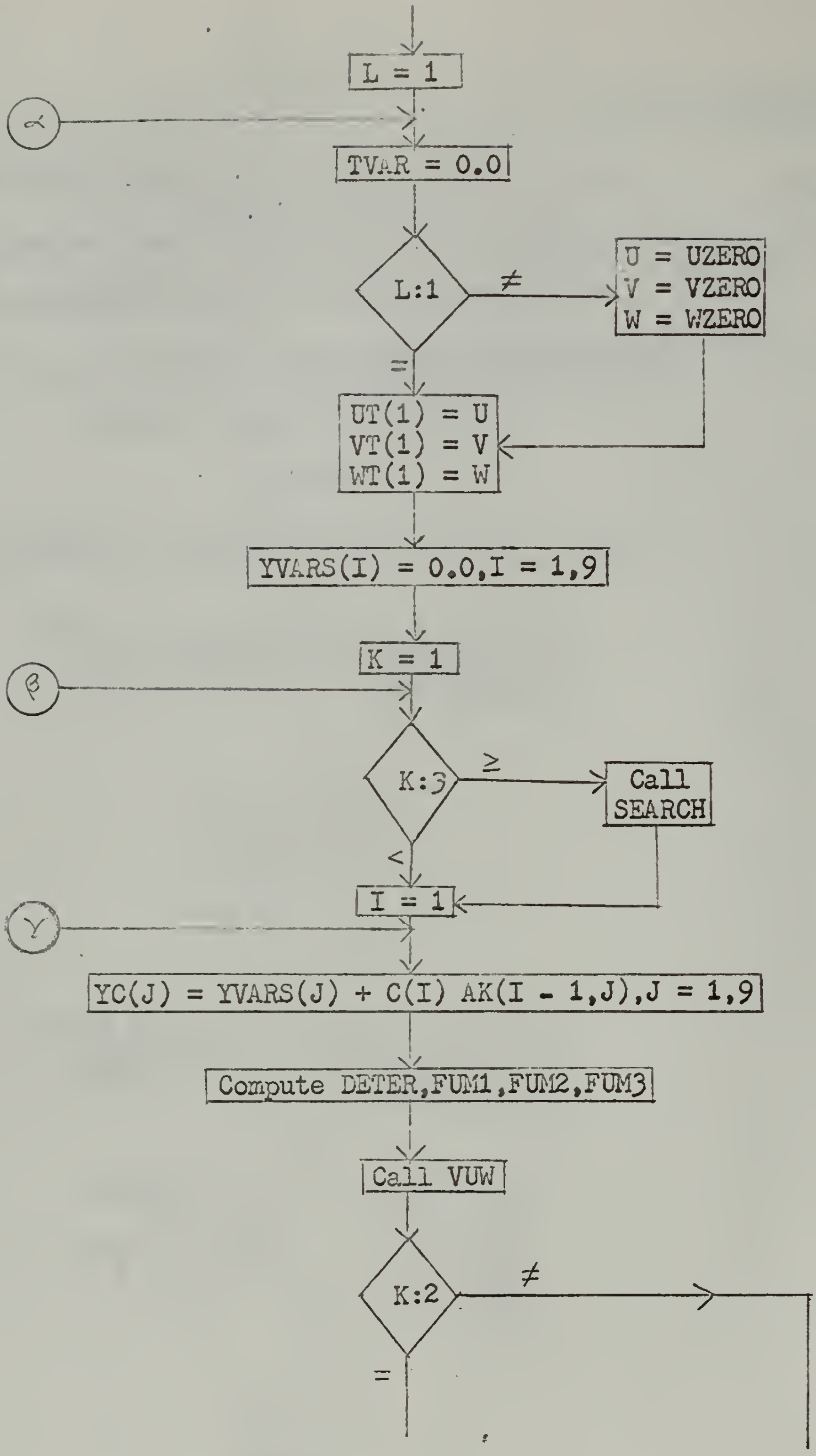
Appendix I

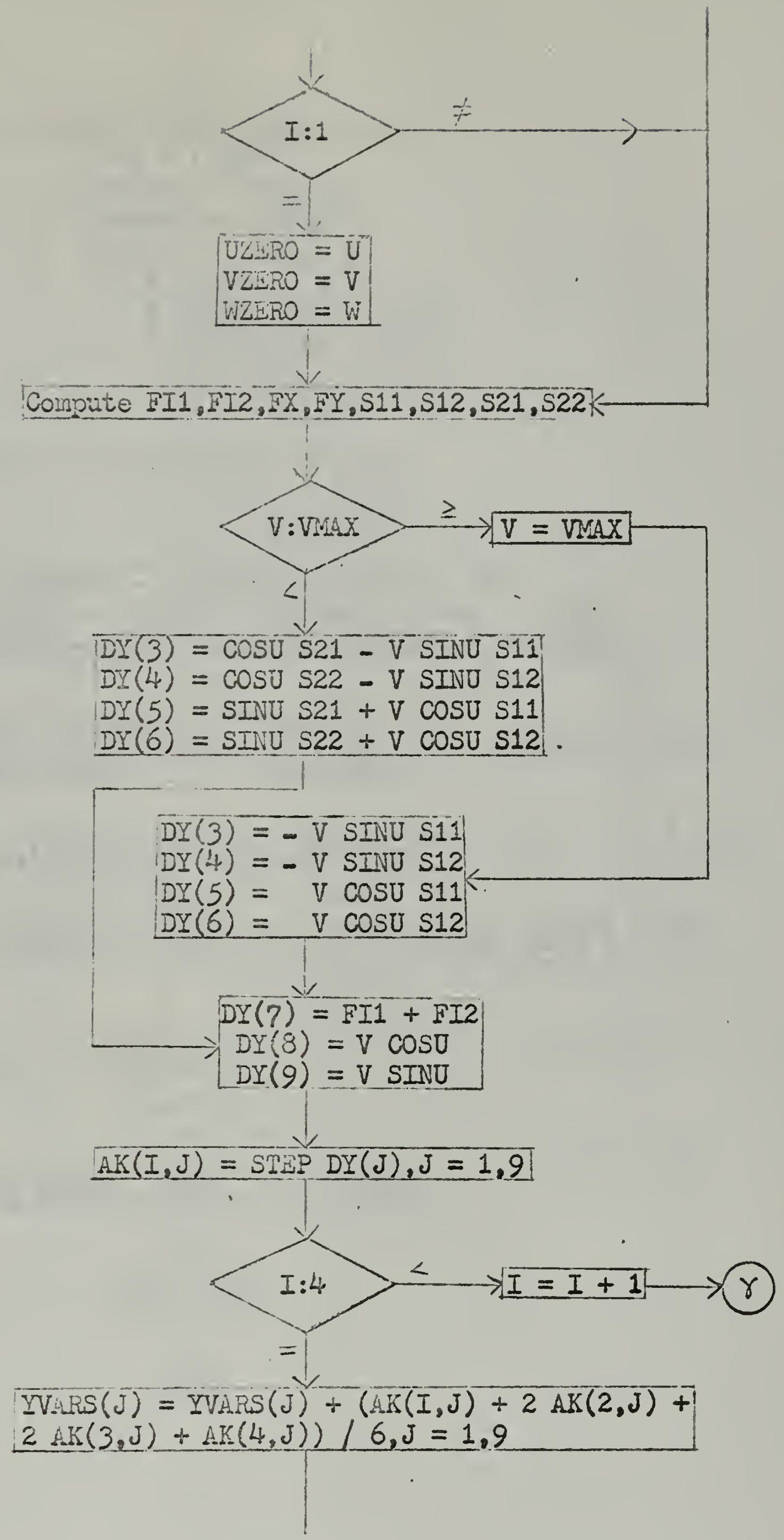
This appendix contains the flow charts and the deck listings for the program of the numerical routine and the subroutines that were used to generate the paths described in the text.

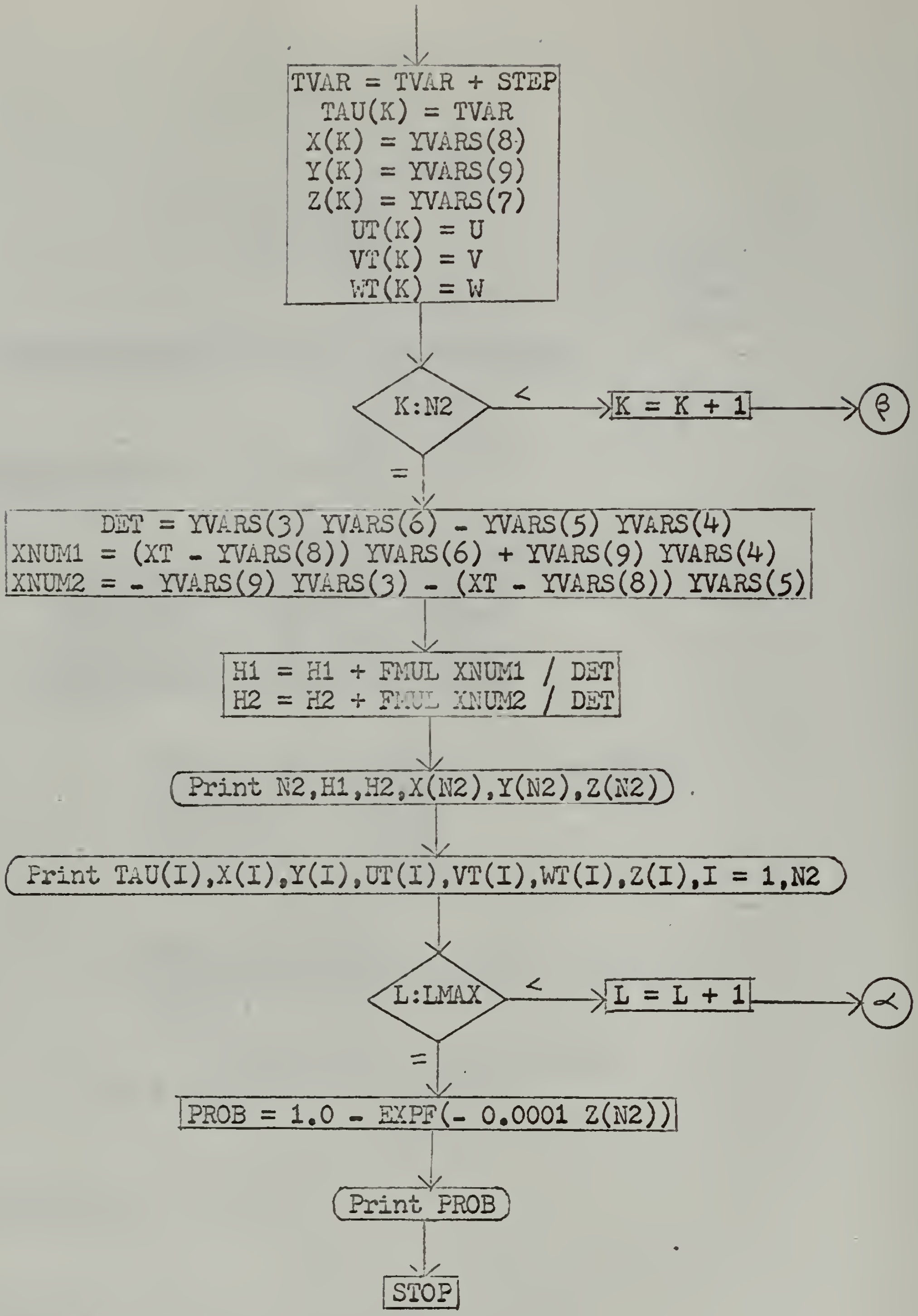
A. Flow chart for the numerical routine.



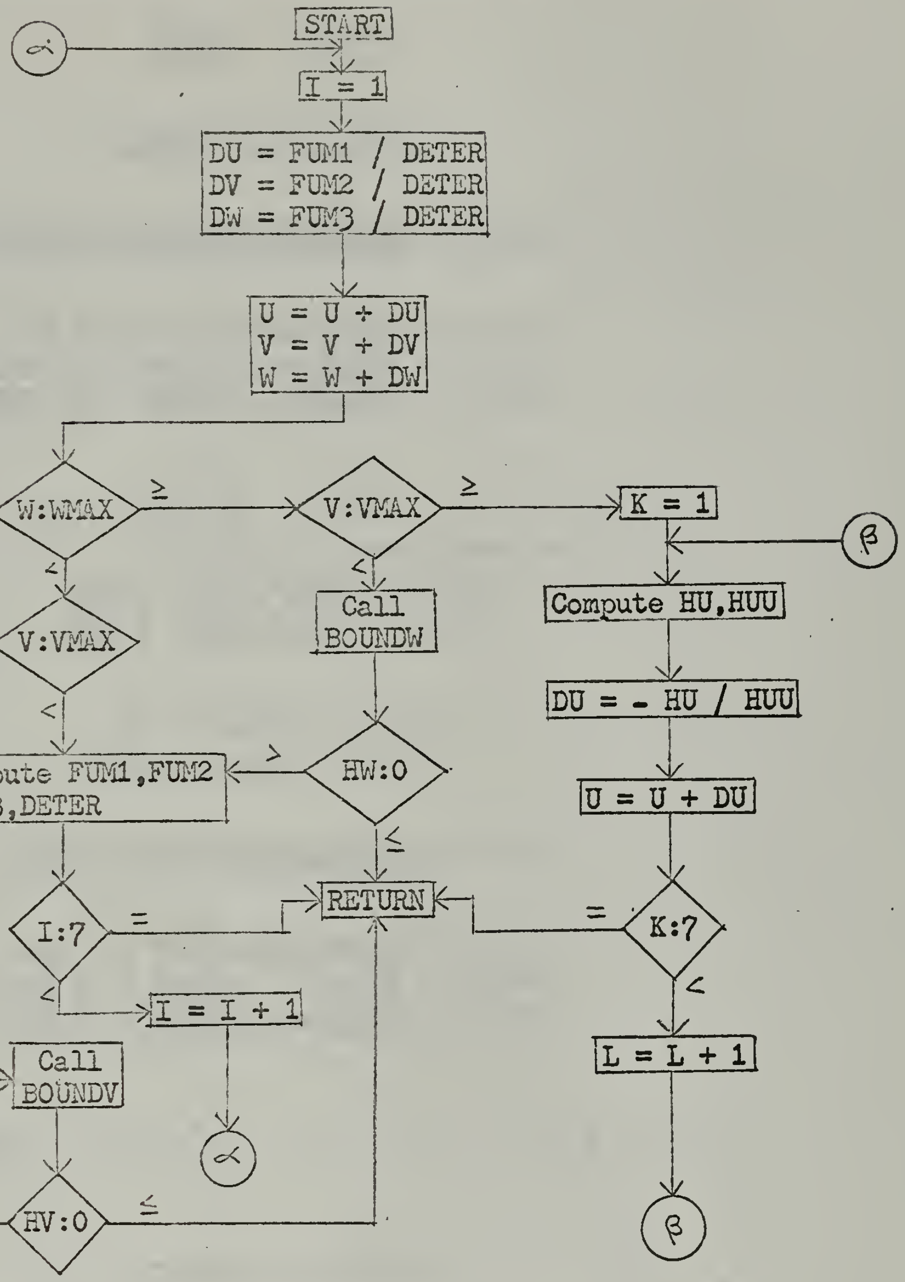
```
START

Dimension YVARS,YC,DY,C,AK,TAU,X,Y,Z,UT,VT,WT

Read XT,T,THETA,A1,B1,C1,D1,S1,C2,
D2,X2,Y2,H1,H2,FAC,FMUL,WO,LMAX,
W1,WMAX,VMAX

Q = THETA / 57.29
  COS = COSF(Q)
  SIN = SINF(Q)
X2 = S1 COS + XT
   Y2 = S1 SIN
   VAV = XT / T

Print XT,T,THETA,A1,B1,C1,D1,S1,C2,
D2,X2,Y2,H1,H2,FAC,FMUL,WO,LMAX,
W1,WMAX,VMAX

 C(1) = 0.0      N1 = T / XSTEP
 C(2) = 0.0         XN1 = N1
 C(3) = 0.0       STEP = T / XN1
 C(4) = 0.0        N2 = N1 + 1
XSTEP = T / FAC

U = - .2
 V = VAV
 W = WO
```


L = 1
α
TVAR = 0.0
L:1
≠
U = UZERO
V = VZERO
W = WZERO
=
UT(1) = U
VT(1) = V
WT(1) = W
YVARS(I) = 0.0,I = 1,9
K = 1
β
K:3
≥
Call SEARCH
<
I = 1
γ
YC(J) = YVARS(J) + C(I) AK(I - 1,J),J = 1,9
Compute DETER,FUM1,FUM2,FUM3
Call VUW
K:2
≠
=

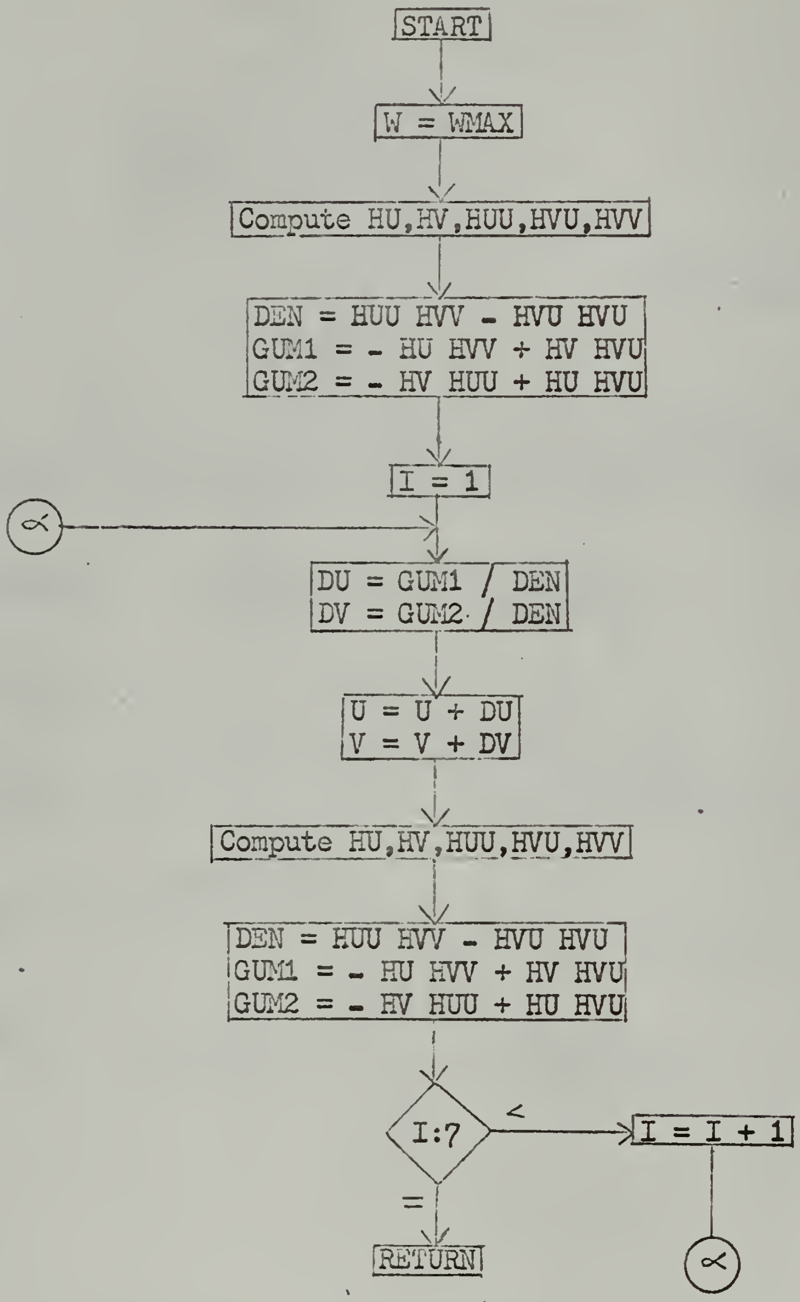

I:1
≠
=
UZERO = U
VZERO = V
WZERO = W
Compute FI1,FI2,FX,FY,S11,S12,S21,S22
V:VMAX
≥
V = VMAX
<
DY(3) = COSU S21 - V SINU S11
DY(4) = COSU S22 - V SINU S12
DY(5) = SINU S21 + V COSU S11
DY(6) = SINU S22 + V COSU S12
DY(3) = - V SINU S11
DY(4) = - V SINU S12
DY(5) = V COSU S11
DY(6) = V COSU S12
DY(7) = FI1 + FI2
DY(8) = V COSU
DY(9) = V SINU
AK(I,J) = STEP DY(J),J = 1,9
I:4
<
I = I + 1
γ
=
YVARS(J) = YVARS(J) + (AK(I,J) + 2 AK(2,J) +
2 AK(3,J) + AK(4,J)) / 6,J = 1,9


TVAR = TVAR + STEP
TAU(K) = TVAR
X(K) = YVARS(8)
Y(K) = YVARS(9)
Z(K) = YVARS(7)
UT(K) = U
VT(K) = V
WT(K) = W
K:N2
<
K = K + 1
β
=
DET = YVARS(3) YVARS(6) - YVARS(5) YVARS(4)
XNUM1 = (XT - YVARS(8)) YVARS(6) + YVARS(9) YVARS(4)
XNUM2 = - YVARS(9) YVARS(3) - (XT - YVARS(8)) YVARS(5)
H1 = H1 + FMUL XNUM1 / DET
H2 = H2 + FMUL XNUM2 / DET
Print N2,H1,H2,X(N2),Y(N2),Z(N2)
Print TAU(I),X(I),Y(I),UT(I),VT(I),WT(I),Z(I),I = 1,N2
L:LMAX
<
L = L + 1
α
=
PROB = 1.0 - EXPF(- 0.0001 Z(N2))
Print PROB
STOP

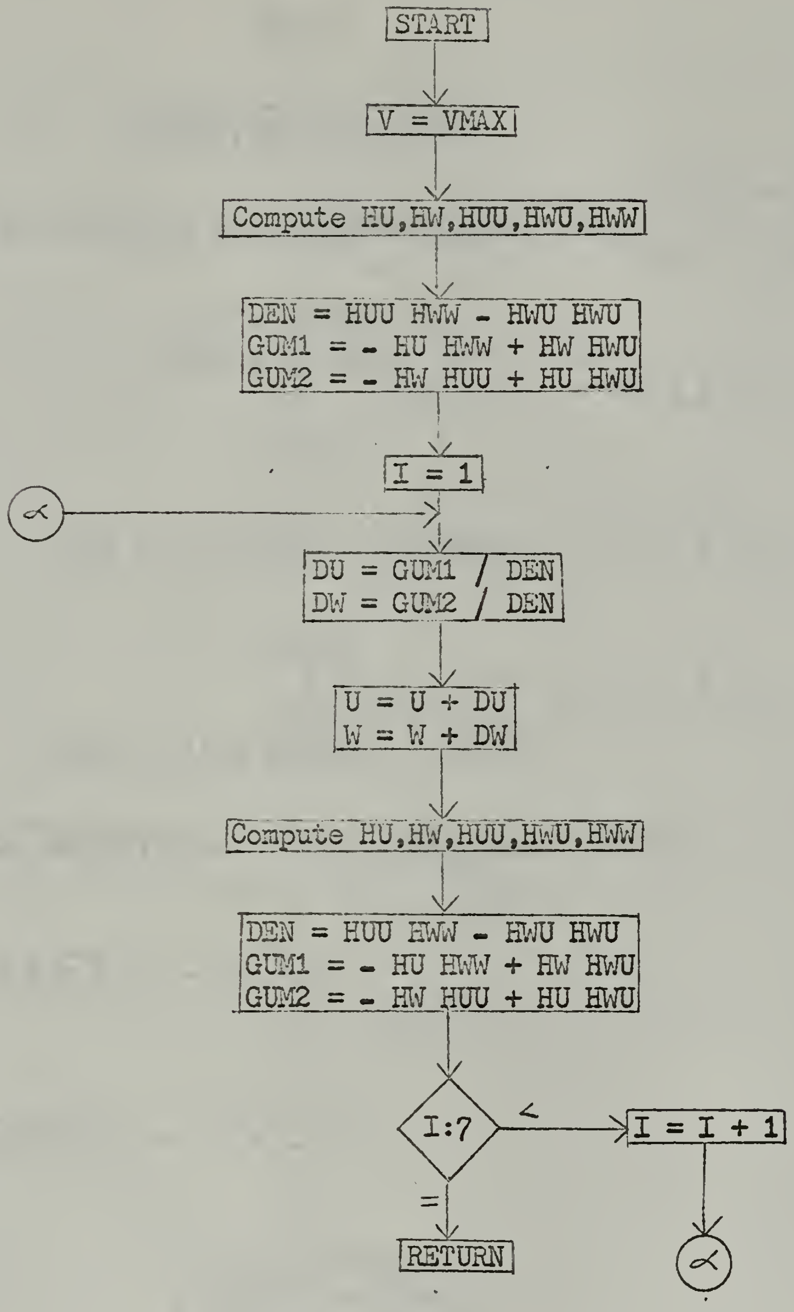
B. Flow chart for subroutine VUW.



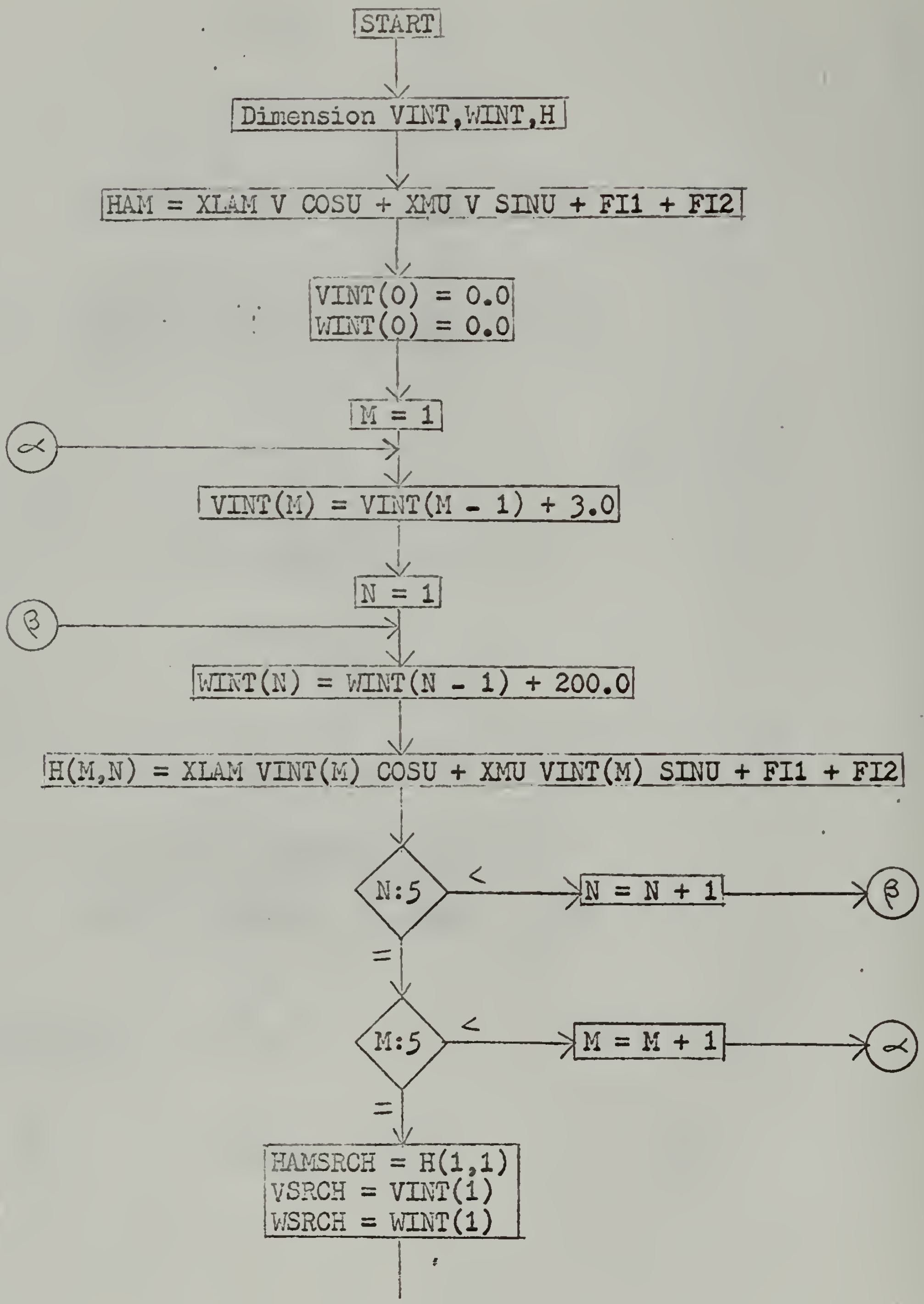
C. Flow chart for subroutine BOUNDW.

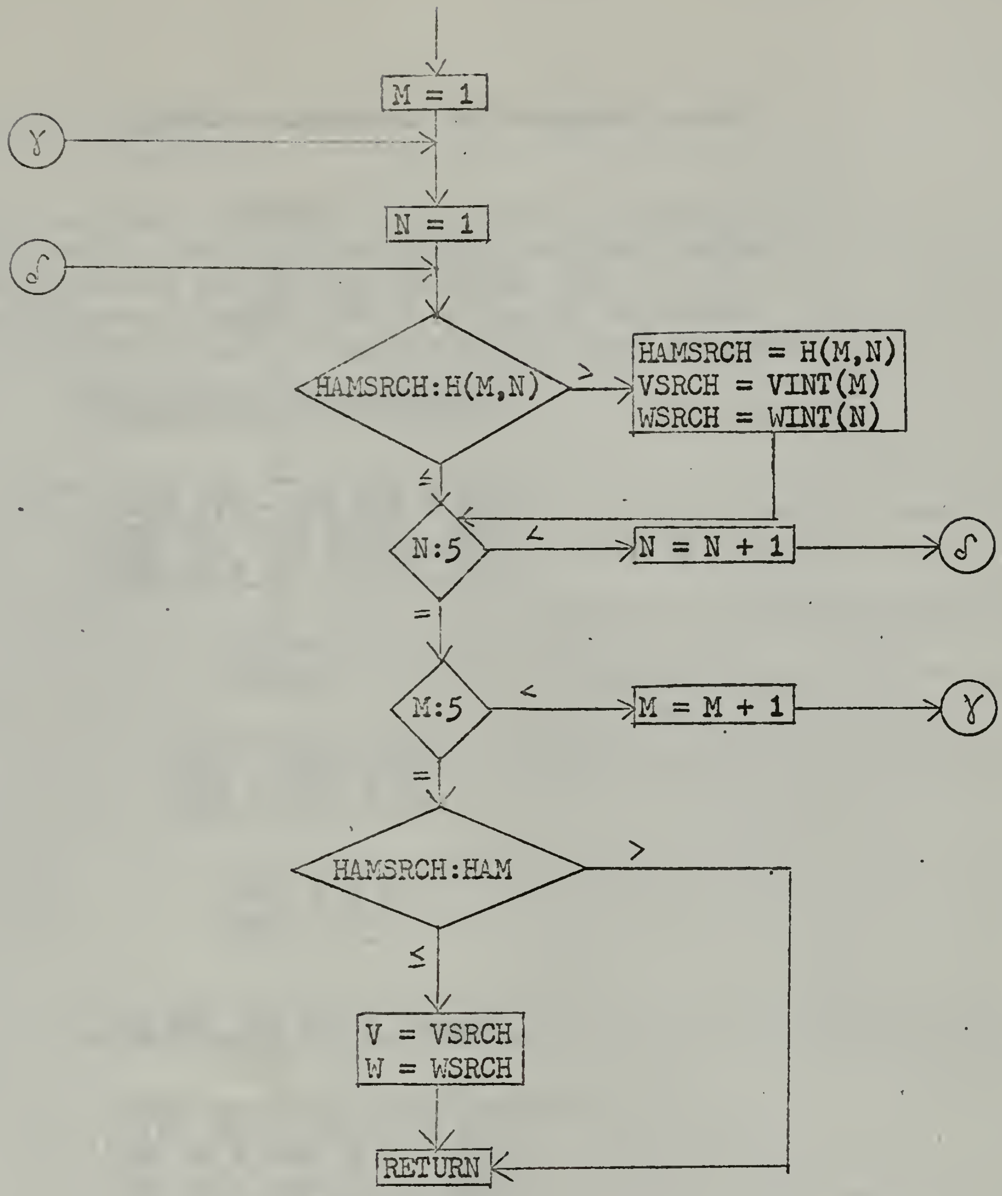


D. Flow chart for subroutine BOUNDV.

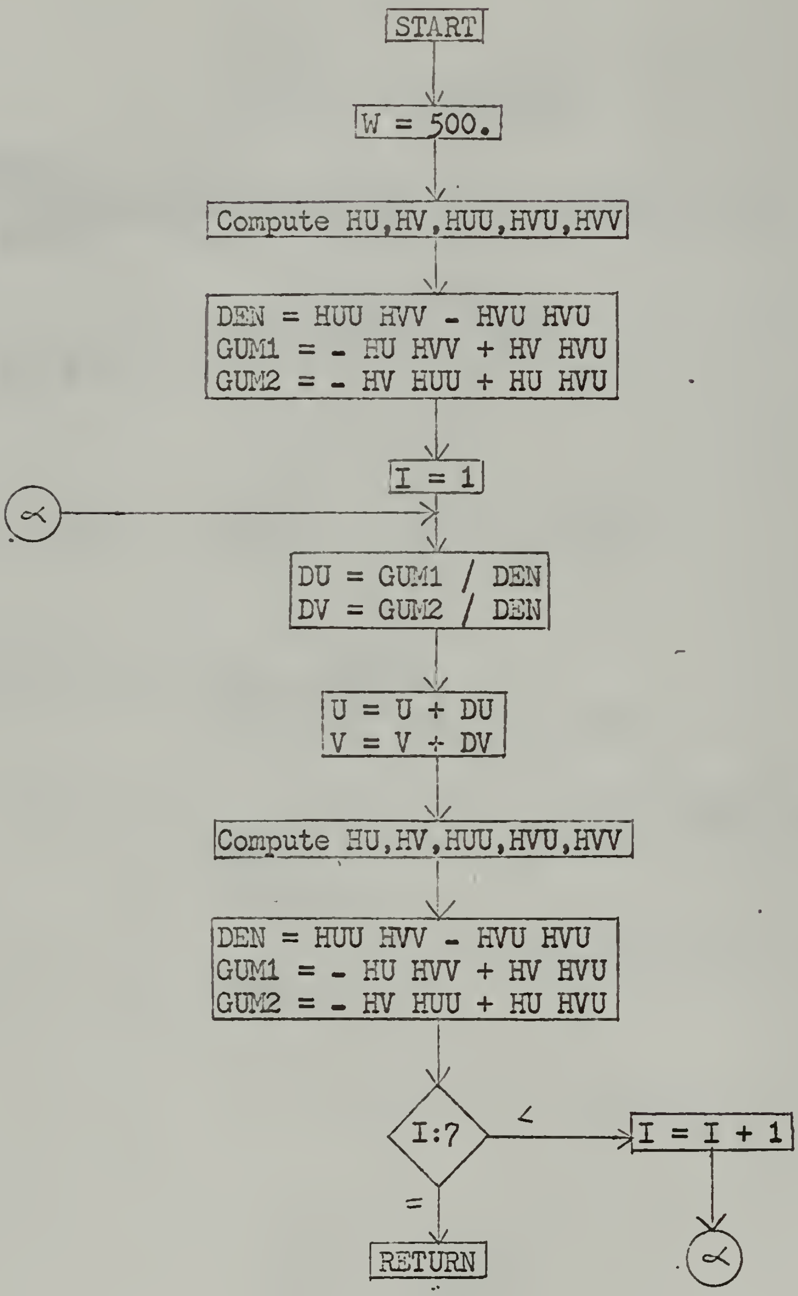


E. Flow chart for subroutine SEARCH.


START
Dimension VINT,WINT,H
HAM = XLAM V COSU + XMU V SINU + FI1 + FI2
VINT(0) = 0.0
WINT(0) = 0.0
M = 1
α
VINT(M) = VINT(M - 1) + 3.0
N = 1
β
WINT(N) = WINT(N - 1) + 200.0
H(M,N) = XLAM VINT(M) COSU + XMU VINT(M) SINU + FI1 + FI2
N:5
<
N = N + 1
β
=
M:5
<
M = M + 1
α
=
HAMSRCH = H(1,1)
VSRCH = VINT(1)
WSRCH = WINT(1)


M = 1
γ
N = 1
δ
HAMSRCH:H(M,N)
>
HAMSRCH = H(M,N)
VSRCH = VINT(M)
WSRCH = WINT(N)
≤
N:5
<
N = N + 1
δ
=
M:5
<
M = M + 1
γ
=
HAMSRCH:HAM
>
≤
V = VSRCH
W = WSRCH
RETURN

F. Flow chart for subroutine WORSI.



G. Fortran Program: Printout of deck of punch cards submitted to computer.

The main numerical routine and various subroutines are given here. The equations in the numerical routine and the first four subroutines as listed here are the ones which gave a corner. The equations in subroutine WORSI are the ones used in generating the three extremals.

```
      PROGRAM SUBROUTE
C   YVARS(1)=LAM3 YVARS(2)=MU3  YVARS(3)=A11  YVARS(4)=A12  YVARS(5)=A21
C   YVARS(6)=A22  YVARS(7)=  Z  YVARS(8)=  X  YVARS(9)=  Y
C   XLAM=H1+LAM3  XMU=H2+MU3
      DIMENSION YVARS(9),YC(9),DY(9),C(4),AK(4,9),TAU(400),X(900),
     +          Y(90 ),Z(400),VT(400),UT(400),WT(400)
      READ 1,XT,T,THETA,A1,B1,C1,D1,S1,C2,D2,X2,Y2,H1,H2,FAC,FMUL,
     +WO,LMAX,W1,WMAX,VMAX
    1 FORMAT (F5.0,2F4.0,F2.0,F3.2,F2.1,F5.4,F4.0,F7.6,F5.4,F6.1,F4.1,
     +    F6.4,F6.0,F4.0,F4.2,F4.0,I2/F4.0,F5.0,F3.0)
      Q=THETA/57.2957 7951
      COS=COSF(Q)
      SIN=SINF(Q)
      X2=S1*COS+XT
      Y2=S1*SIN
      VAV=XT/T
      PRINT 2
    2 FORMAT(1H02HXT4X1HT3X5HTHETA2X2HA12X2HB12X2HC12X2HD12X2HS12X4HWMAX
     +1X4HVMAX)
      PRINT 3,XT,T,THETA,A1,B1,C1,D1,S1,WMAX,VMAX
    3 FORMAT (3F5.0,F5.1,F4.2,F4.1,F5.4,F5.0,F5.0,F3.0/)
      PRINT 4
    4 FORMAT(3H C24X2HD24X2HX24X2HY25X2HH17X2HH23X3HFAC2X4HLMAX
     +3X4HFMUL2X2HWO5X2HW1)
      PRINT 5,C2,D2,X2,Y2,H1,H2,FAC,LMAX,FMUL,WO,W1
    5 FORMAT (F6.5,F5.1,F6.1,F4.1,2F9.5,F6.0,I3,F8.2,F6.0,F6.0/)
C   DEFINE RUNGE-KUTTA CONSTANTS.
      C(1)=0.0
      C(2)=0.5
      C(3)=0.5
      C(4)=1.0
      X(1)=0.0
      Y(1)=0.0
      TAU(1)=0.0
C  COMPUTE TIME STEP LENGTH.
      XSTEP=T/FAC
      N1=T/XSTEP
      XN1=N1
      STEP=T/XN1
      N2=N1+1
C  GUESS INITIAL VALUES FOR U, V, AND W BEFORE FIRST ITERATION
      U=-.9
      V=VAV
      W=WO
C  ENTER LOOP FOR GENERATING CORRECTIONS TO H1,H2.
      DO 14 L=1,LMAX
      TVAR=0.0
C  GUESS INITIAL VALUES FOR U,V,W, AFTER FIRST ITERATION
      IF (L-1) 198,199,198
  198 U=UZERO
      V=VZERO
      W=WZERO
  199 UT(1)=U
      VT(1)=V
      WT(1)=W
      DO 6 I=1,9
    6 YVARS(I)=0.0
C  ENTER LOOP FOR COMPUTING THE PATH FOR EACH TIME STEP.
      DO 11 K=2,N2
      IF(K-3) 901,902,902
  902 CALL SEARCH(A1,B1,C1,D1,FI1,FI2,XLAM,XMU,COSU,U,V,W,SINU,WO,
     +G1,G2,W1,COSUMQ)
C  ENTER LOOP FOR RUNGE-KUTTA INTEGRATION.
  901 DO 88 I=1,4
```

```
      DO 7 J=1,9
    7 YC(J)=YVARS(J)+C(I)*AK(I-1,J)
      XLAM=H1+YC(1)
      XMU=H2+YC(2)
      XD=(XT-YC(8))*COS+S1-YC(9)*SIN
      FI1=EXPF(-(XD*LOGF(2.)/1000.))
      FI2=C2+D2*EXPF(-((YC(8)-X2)**2+(YC(9)-Y2)**2)*LOGF(2.)/100000.)
      COSU=COSF(U)
      SINU=SINF(U)
      COSUMQ=COSF(U-Q)
      SINUMQ=SINF(U-Q)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      G7=1.+W*W/(W1*W1)
      P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
      P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
      P9=FI2*(-2./(W1*W1*G7*G7)+8.*W*W/(W1*W1*W1*W1*G7*G7*G7))
      HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
      HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
      HWU=G3*(.25*(SINUMQ))*G4
      HWV=G4*(.01)*FI1*(2.*B1*V)*(1.-.25*(COSUMQ))
      HWW=   G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
      P5=HVV*HWW-HWV*HWV
      P6=(-HV)*HWW+HW*HWV
      P7=HVV*(-HW)+HWV*HV
      DETER=HUU*P5-HVU*(HVU*HWW-HWU*HWV)+HWU*(HVU*HWV-HWU*HVV)
      FUM1=(-HU)*P5-HVU*P6+HWU*((-HV)*HWV+HW*HVV)
      FUM2=HUU*P6+HU*(HVU*HWW-HWU*HWV)+HWU*(HVU*(-HW)+HWU*HV)
      FUM3=HUU*P7-HVU*(HVU*(-HW)+HWU*HV)-HU*(HVU*HWV-HWU*HVV)
C  CALL SUBROUTINE FOR CORRECTIONS TO U,V,W.
      CALL VUW(DETER,FUM1,FUM2,FUM3,XLAM,XMU,FI1,FI2,A1,B1,C1,D1,WO,
     +W1,WMAX,VMAX,U,V,W,Q)
C  FIV=COEFF OF FI1      F2V=COEFF OF FI2
      IF(K-2) 195,297,195
  297 IF (I-1) 195,196,195
  196 UZERO=U
      VZERO=V
      WZERO=W
  195 XD=(XT-YC(8))*COS+S1-YC(9)*SIN
      FI1=EXPF(-(XD*LOGF(2.)/1000.))
      FI2=C2+D2*EXPF(-((YC(8)-X2)**2+(YC(9)-Y2)**2)*LOGF(2.)/100000.)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      G7=1.+(W/W1)**2
      COSU=COSF(U)
      SINU=SINF(U)
      SINUMQ=SINF(U-Q)
      COSUMQ=COSF(U-Q)
      P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
      P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G2
      P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
      P9=FI2*(-2./(W1*W1*G7*G7)+8.*W*W/(W1*W1*W1*W1*G7*G7*G7))
      HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
      HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
      HWU=G3*(.25*(SINUMQ))*G4
      HWV=G4*(.01)*FI1*(2.*B1*V)*(1.-.25*(COSUMQ))
      HWW=  G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
      F1V=(A1+B1*V*V)*(.01)*(1.-.25*COSUMQ)*G1/G2
      F2V=1./(1.+(W*W/(W1*W1)))
      FY=F1V*FI1*SIN*LOGF(2.)/1000.+F2V*(-D2*EXPF(-((YC(8)-X2)**2+
     +(YC(9)-Y2)**2)*LOGF(2.)/100000. )*(2.*(YC(9)-Y2)*LOGF(2.)/100000.
     +))
      FX=F1V*FI1*SIN*LOGF(2.)/1000.+F2V*(-D2*EXPF(-((YC(8)-X2)**2+
     +(YC(9)-Y2)**2)*LOGF(2.)/100000. )*(2.*(YC(8)-X2)*LOGF(2.)/100000.
     +))
      DENOM=HVU*(HVU*HWW-HWU*HWV)-HUU*(HVV*HWW-HWV*HWV)+HWU*(HVV*HWU-
     +HWV*HVU)
      S11=((HVV*HWW-HWV*HWV)*(-V*SINU)+(HVU*HWW-HWV*HWU)*(-COSU))/DENOM
      S12=((HVV*HWW-HWV*HWV)*(V*COSU)+(HVU*HWW-HWV*HWU)*(-SINU))/DENOM
      S21=((HVU*HWW-HWU*HWV)*(V*SINU)+(HUU*HWW-HWU*HWU)*COSU)/DENOM
      S22=((HVU*HWW-HWU*HWV)*((-V)*COSU)+(HUU*HWW-HWU*HWU)*SINU)/DENOM
      DY(1)=-FX
      DY(2)=-FY
      IF(V-VMAX)118,119,119
  118 DY(3)=COSU*S21-V*SINU*S11
      DY(4)=COSU*S22-V*SINU*S12
      DY(5)=SINU*S21+V*COSU*S11
      DY(6)=SINU*S22+V*COSU*S12
      GO TO 120
  119 V=VMAX
      DY(3)=         -V*SINU*S11
      DY(4)=         -V*SINU*S12
      DY(5)=          V*COSU*S11
      DY(6)=          V*COSU*S12
  120 DY(7)=FI1*F1V+FI2*F2V
      DY(8)=V*COSU
      DY(9)=V*SINU
      DO 8 J=1,9
    8 AK(I,J)=STEP*DY(J)
   88 CONTINUE
      DO 9 J=1,9
  9 YVARS(J)=YVARS(J)+(AK(1,J)+2.*AK(2,J)+2.*AK(3,J)+AK(4,J))/6.
    TVAR=TVAR+STEP
    TAU(K)=TVAR
    X(K)=YVARS(8)
    Y(K)=YVARS(9)
    Z(K)=YVARS(7)
    UT(K)=U
    VT(K)=V
    WT(K)=W
 11 CONTINUE
    DET=YVARS(3)*YVARS(6)-YVARS(5)*YVARS(4)
    XNUM1=(XT-YVARS(8))*YVARS(6)+YVARS(9)*YVARS(4)
    XNUM2=-YVARS(9)*YVARS(3)-(XT-YVARS(8))*YVARS(5)
    H1=H1+     FMUL*XNUM1/DET
    H2=H2+     FMUL*XNUM2/DET
    PRINT 12
 12 FORMAT (2X2HN25X2HH17X2HH25X5HX(N2)4X5HY(N2)6X5HZ(N2))
    PRINT 13,N2,H1,H2,X(N2),Y(N2),Z(N2)
 13 FORMAT (I4,2F9.5,2F9.1,E13.5/)
 18 PRINT 19
 19 FORMAT (1H03X3HTAU6X1HX7X1HY7X1HU5X1HV5X1HW10X1HZ/)
    PRINT 20,(TAU(I),X(I),Y(I),UT(I),VT(I),WT(I),Z(I),I=1,N2)
 20 FORMAT (F8.2,2F8.1,3F6.1,E13.5)
 14 CONTINUE
    PROB=1.0-EXPF(-0.0001*Z(N2))
    PRINT 21,PROB
 21 FORMAT (1H02X5HPROB=E13.5/)
    STOP
    END


    SUBROUTINE VUW(DETER,FUM1,FUM2,FUM3,XLAM,XMU,FI1,FI2,A1,B1,C1,D1,
   +WO,W1,WMAX,VMAX,U,V,W,Q)
    DO 5 I=1,7
    DU=FUM1/DETER
```

```
      DV=FUM2/DETER
      DW=FUM3/DETER
      U=U+DU
      V=V+DV
      W=W+DW
      IF(WMAX -W) 390,390,392
  390 IF(VMAX-V) 1000,1000,1001
  392 IF(VMAX-V) 3000,3000,3001
 3000 CALL BOUNDV(U,V,W,XLAM,XMU,FI1,A1,B1,C1,D1,WO,FI2,Q,VMAX,HV,W1)
      IF(HV-0.0) 4000,4000,3001
 4000 RETURN
 1000 DO 2000 I=1,5
      V=VMAX
      W=WMAX
      COSU=COSF(U)
      SINU=SINF(U)
      COSUMQ=COSF(U-Q)
      SINUMQ=SINF(U-Q)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      DU=-HU/HUU
      U=U+DU
 2000 CONTINUE
      RETURN
 1001 CALL BOUNDW(U,V,W,XLAM,XMU,FI1,A1,B1,C1,D1,WO,FI2,W1,Q,WMAX,HW)
      IF(HW-0.0) 5000,5000,3001
 5000 RETURN
 3001 COSU=COSF(U)
      SINU=SINF(U)
      COSUMQ=COSF(U-Q)
      SINUMQ=SINF(U-Q)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      G7=1.+W*W/(W1*W1)
      P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
      P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
      P9=FI2*(-2./(W1*W1*G7*G7)+8.*W*W/(W1*W1*W1*W1*G7*G7*G7))
      HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
      HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
      HWU=G3*(.25*(SINUMQ))*G4
      HWV=G4*(.01)*FI1*(2.*B1*V)*(1.-.25*(COSUMQ))
      HWW=  G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
      P5=HVV*HWW-HWV*HWV
      P6=(-HV)*HWW+HW*HWV
      P7=HVV*(-HW)+HWV*HV
      DETER=HUU*P5-HVU*(HVU*HWW-HWU*HWV)+HWU*(HVU*HWV-HWU*HVV)
      FUM1=(-HU)*P5-HVU*P6+HWU*((-HV)*HWV+HW*HVV)
      FUM2=HUU*P6+HU*(HVU*HWW-HWU*HWV)+HWU*(HVU*(-HW)+HWU*HV)
      FUM3=HUU*P7-HVU*(HVU*(-HW)+HWU*HV)-HU*(HVU*HWV-HWU*HVV)
    5 CONTINUE
      RETURN
      END
```

```
 SUBROUTINE BOUNDW(U,V,W,XLAM,XMU,FI1,A1,B1,C1,D1,WO,FI2,W1,Q,WMAX,
+HW)
 W=WMAX
 COSU=COSF(U)
 SINU=SINF(U)
 COSUMQ=COSF(U-Q)
 SINUMQ=SINF(U-Q)
 G1=C1+D1*(W-WO)*(W-WO)
 G2=1.+D1*(W-WO)*(W-WO)
 G3=(.01)*FI1*(A1+B1*V*V)
 G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
 G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
 G7=1.+(W/W1)**2
 HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
 HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
 HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
 HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
 HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
 HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
 DEN=HVU*HVU-HUU*HVV
 GUM1=-HV*HVU+HU*HVV
 GUM2=HVU*(-HU)+HV*HUU
 DO 981  I=1,7
 DU=GUM1/DEN
 DV=GUM2/DEN
 U=U+DU
 V=V+DV
 COSU=COSF(U)
 SINU=SINF(U)
 COSUMQ=COSF(U-Q)
 SINUMQ=SINF(U-Q)
 G1=C1+D1*(W-WO)*(W-WO)
 G2=1.+D1*(W-WO)*(W-WO)
 G3=(.01)*FI1*(A1+B1*V*V)
 G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
 G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
```

```
      HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
      HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      DEN=HVU*HVU-HUU*HVV
      GUM1=-HV*HVU+HU*HVV
      GUM2=HVU*(-HU)+HV*HUU
  981 CONTINUE
      RETURN
      END


      SUBROUTINE BOUNDV(U,V,W,XLAM,XMU,FI1,A1,B1,C1,D1,WO,FI2,Q,VMAX,
     +W1)
      V=VMAX
      COSU=COSF(U)
      SINU=SINF(U)
      COSUMQ=COSF(U-Q)
      SINUMQ=SINF(U-Q)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      G7=1.+W*W/(W1*W1)
      P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
      P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G
      P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
      P9=FI2*(-2./(W1*W1*G7*G7)+8.*W*W/(W1*W1*W1*W1*G7*G7*G7))
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
      HWU=G3*(.25*(SINUMQ))*G4
      HWW=  G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
      DEN=HWU*HWU-HUU*HWW
      GUM1=-HW*HWU+HU*HWW
      GUM2=-HU*HWU+HW*HUU
      DO 981 I=1,7
      DU=GUM1/DEN
      DW=GUM2/DEN
      U=U+DU
      W=W+DW
      COSU=COSF(U)
      SINU=SINF(U)
      COSUMQ=COSF(U-Q)
      SINUMQ=SINF(U-Q)
      G1=C1+D1*(W-WO)*(W-WO)
      G2=1.+D1*(W-WO)*(W-WO)
      G3=(.01)*FI1*(A1+B1*V*V)
      G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
      G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
      G7=1.+W*W/(W1*W1)
      P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
      P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
      P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
      P9=FI2*(-2./(W1*W1*G7*G7)+8.*W*W/(W1*W1*W1*W1*G7*G7*G7))
      HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
      HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
      HW=G3*(1.-.25*(COSUMQ))*G4+FI2*(-2.*(W/W1)*(1./W1))/(G7*G7)
      HWU=G3*(.25*(SINUMQ))*G4
      HWW=  G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
      DEN=HWU*HWU-HUU*HWW
      GUM1=-HW*HWU+HU*HWW
      GUM2=-HU*HWU+HW*HUU
  981 CONTINUE
      RETURN
      END
      SUBROUTINE SEARCH(A1,B1,C1,D1,FI1,FI2,XLAM,XMU,COSU,U,V,W,SINU,
     +WO,G1,G2,W1,COSUMQ)
      DIMENSION VINT(10),WINT(10),H(50,50)
      F1V=(A1+B1*V*V)*(.01)*(1.-.25*COSUMQ)*G1/G2
      F2V=1./(1.+(W*W/(W1*W1)))
      HAM     =XLAM*V*COSU+XMU*V*SINU+F1V*FI1+F2V*FI2
      VINT(0)=0.0
      WINT(0)=0.0
      DO 131 M=1,5
      VINT(M)=VINT(M-1)+3.0
      DO 122 N=1,5
      WINT(N)=WINT(N-1)+200.0
      G1=C1+D1*(WINT(N)-WO)*(WINT(N)-WO)
      G2=1.+D1*(WINT(N)-WO)*(WINT(N)-WO)
      F1V=(A1+B1*VINT(M)*VINT(M))*(.01)*(1.-.25*COSUMQ)*G1/G2
      F2V=1./(1.+WINT(N)*WINT(N)/(W1*W1))
      H(M,N)=XLAM*VINT(M)*COSU+XMU*VINT(M)*SINU+F1V*FI1+F2V*FI2
  122 CONTINUE
  131 CONTINUE
      HAMSRCH=H(1,1)
      VSRCH=VINT(1)
      WSRCH=WINT(1)
      DO 121 M=1,5
      DO 123 N=1,5
      IF (HAMSRCH-H(M,N)) 123,123,111
  111 HAMSRCH=H(M,N)
      VSRCH=VINT(M)
      WSRCH=WINT(N)
  123 CONTINUE
  121 CONTINUE
      IF (HAMSRCH-HAM) 1,2;2
    1 V=VSRCH
      W=WSRCH
    2 RETURN
      END
```

```
SUBROUTINE WORSI (U,V,W,XLAM,XMU,FI1,A1,B1,C1,D1,WO,FI2,W1,Q)
W=500.
SINU=SINF(U)
COSUMQ=COSF(U-Q)
SINUMQ=SINF(U-Q)
G1=C1+D1*(W-WO)*(W-WO)
G2=1.+D1*(W-WO)*(W-WO)
G3=(.01)*FI1*(A1+B1*V*V)
G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
G7=1.+W/WO
P1=((1.+D1*(W-WO)*(W-WO))*2.*D1+4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
P2=(-(8.*D1*(W-WO)*(W-WO))*D1)/(G2*G2)
P3= (-(C1+D1*(W-WO)*(W-WO))*2.*D1-4.*D1*(W-WO)*D1*(W-WO))/(G2*G2)
P4= (G1*8.*D1*(W-WO)*D1*(W-WO))/(G2*G2*G2)
P9=FI2*(((-G7*(.1/WO)+G6*(1./WO))*2.*(1./WO))/(G7*G7*G7))
HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
HW=G3*(1.-.25*(COSUMQ))*G4+(FI2*((G7*(.1/WO))-G6*(1./WO)))/(G7*G7)
HWW=  G3*((1.-.25*COSUMQ)*(P1+P2+P3+P4))+P9
DEN=HVU*HVU-HUU*HVV
GUM1=-HV*HVU+HU*HVV
GUM2=HVU*(-HU)+HV*HUU
DO 981 I=1,7
DU=GUM1/DEN
DV=GUM2/DEN
U=U+DU
V=V+DV
COSU=COSF(U)
SINU=SINF(U)
```

```
    COSUMQ=COSF(U-Q)
    SINUMQ=SINF(U-Q)
    G1=C1+D1*(W-WO)*(W-WO)
    G2=1.+D1*(W-WO)*(W-WO)
    G3=(.01)*FI1*(A1+B1*V*V)
    G4=(-G1*(2.*D1*(W-WO))+G2*(2.*D1*(W-WO)))/(G2*G2)
    G5=(C1+D1*(W-WO)*(W-WO))/(1.+D1*(W-WO)*(W-WO))
    F1V=(A1+B1*V*V)*(.01)*(1.-.25*COSUMQ)*G1/G2
    F2V=(1.+.1*W/WO)/(1.+W/WO)
    HV=XLAM*COSU+XMU*SINU+(.01)*FI1*(2.*B1*V)*G5*(1.-.25*COSUMQ)
    HU=-XLAM*V*SINU+XMU*V*COSU+G3*G5*(.25*SINUMQ)
    HVV=(.01)*2.*B1*FI1*G5*(1.-.25*COSUMQ)
    HVU=-XLAM*SINU+XMU*COSU+G5*.01*FI1*2.*B1*V*(.25*SINUMQ)
    HUU=-XLAM*V*COSU-XMU*V*SINU+G3*G5*(.25*COSUMQ)
    DEN=HVU*HVU-HUU*HVV
    GUM1=-HV*HVU+HU*HVV
    GUM2=HVU*(-HU)+HV*HUU
981 CONTINUE
    RETURN
    END
```

Appendix II

In this appendix additional information about the paths described in section 7 and the path discussed in section 8 will be given. For the first three paths the position (x,y) and the controls u,v,w for the submarine will be given at twelve-hour intervals. This information will be given for Path IV and in addition the submarine's location and controls will be included for the time steps immediately preceeding and following the corner.

Path I

Time	x	y	u	v	w
0.0	0.0	0.0	- 0.7	10.7	207.8
12.0	104.5	- 74.6	- 0.6	10.7	207.8
24.0	214.8	- 140.4	- 0.5	10.7	207.8
36.0	330.0	- 197.2	- 0.4	10.7	207.8
48.0	449.2	- 244.8	- 0.3	10.7	207.7
60.0	571.5	- 283.1	- 0.3	10.7	207.5
72.0	696.1	- 312.5	- 0.2	10.7	207.3
84.0	822.3	- 333.0	- 0.1	10.6	207.1
96.0	949.3	- 345.1	- 0.1	10.6	206.8

Time	x	Y	u	V	W
108.0	1076.5	- 349.2	0.0	10.6	206.5
120.0	1203.5	- 345.7	0.1	10.6	206.2
132.0	1329.8	- 335.1	0.1	10.5	205.9
144.0	1455.0	- 318.0	0.2	10.5	205.5
156.0	1578.9	- 294.8	0.2	10.5	205.2
168.0	1701.3	- 266.0	0.3	10.5	204.9
180.0	1822.0	- 232.1	0.3	10.4	204.6
192.0	1941.0	- 193.6	0.3	10.4	204.3
204.0	2058.2	- 150.8	0.4	10.4	204.0
216.0	2173.6	- 104.2	0.4	10.4	203.7
228.0	2287.2	- 54.1	0.4	10.3	203.5
240.0	2400.0	0.0	0.5	11.1	203.7

Path II

Time	x	Y	u	V	W
0.0	0.0	0.0	- 0.6	10.1	1000.0
12.0	102.4	- 66.1	- 0.5	10.2	1000.0
24.0	210.1	- 124.3	- 0.5	10.2	1000.0
36.0	322.4	- 174.3	- 0.4	10.3	1000.0
48.0	438.6	- 216.3	- 0.3	10.3	1000.0
60.0	558.0	- 250.4	- 0.2	10.4	1000.0
72.0	680.0	- 276.7	- 0.2	10.4	1000.0

Time	x	y	u	v	w
84.0	804.0	- 295.6	- 0.1	10.5	1000.0
96.0	929.4	- 307.4	- 0.1	10.5	1000.0
108.0	1055.7	- 312.3	0.0	10.5	1000.0
120.0	1182.4	- 310.9	0.0	10.6	1000.0
132.0	1309.2	- 303.7	0.1	10.6	1000.0
144.0	1435.7	- 290.9	0.1	10.6	1000.0
156.0	1561.7	- 273.2	0.2	10.6	1000.0
168.0	1686.9	- 250.8	0.2	10.6	1000.0
180.0	1811.2	- 224.1	0.2	10.6	1000.0
192.0	1934.5	- 193.7	0.3	10.6	1000.0
204.0	2056.6	- 159.8	0.3	10.6	1000.0
216.0	2177.6	- 122.7	0.3	10.5	1000.0
228.0	2290.2	- 64.0	0.5	10.6	705.2
240.0	2400.0	0.0	0.5	10.7	650.1

Path III

Time	x	y	u	v	w
0.0	0.0	0.0	- 0.7	10.8	527.4
12.0	106.1	- 75.4	- 0.6	10.9	526.4
24.0	218.1	- 141.9	- 0.5	10.9	526.6
36.0	334.9	- 199.1	- 0.4	10.8	528.1
48.0	455.8	- 246.9	- 0.3	10.8	530.7

Time	x	y	u	v	w
60.0	579.6	- 285.4	- 0.3	10.8	534.6
72.0	705.5	- 314.6	- 0.2	10.8	539.6
84.0	832.8	- 334.8	- 0.1	10.7	545.9
96.0	960.6	- 346.5	- 0.1	10.7	553.3
108.0	1088.3	- 350.1	0.0	10.6	561.9
120.0	1215.4	- 346.1	0.1	10.6	571.7
132.0	1341.5	- 335.0	0.1	10.5	582.7
144.0	1466.3	- 317.4	0.2	10.5	594.9
156.0	1589.5	- 293.9	0.2	10.4	608.3
168.0	1710.9	- 264.8	0.3	10.4	622.9
180.0	1830.5	- 230.7	0.3	10.3	638.7
192.0	1948.1	- 192.1	0.3	10.3	655.8
204.0	2063.8	- 149.4	0.4	10.3	674.2
216.0	2177.6	- 103.0	0.4	10.2	693.9
228.0	2289.6	- 53.2	0.4	10.2	715.0
240.0	2400.1	0.0	0.5	10.4	677.9

Path IV

Time	x	y	u	v	w
0.0	0.0	0.0	- 1.1	11.0	200.8
12.0	70.2	- 111.9	- 1.0	11.0	200.8

Time	x	y	u	v	w
24.0	149.6	- 218.0	- 0.9	11.1	200.8
36.0	237.7	- 317.5	- 0.8	11.1	200.8
48.0	334.1	- 409.4	- 0.7	11.1	200.8
60.0	437.9	- 493.1	- 0.6	11.1	200.8
72.0	548.4	- 568.0	- 0.6	11.1	200.9
84.0	664.6	- 633.6	- 0.5	11.1	200.8
96.0	785.7	- 689.6	- 0.4	11.1	200.8
108.0	910.5	- 736.0	- 0.3	11.1	200.8
120.0	1038.3	- 772.7	- 0.2	11.1	200.8
132.0	1168.3	- 799.8	- 0.2	11.1	200.8
144.0	1302.0	- 816.3	- 0.1	11.5	201.1
156.0	1453.5	- 811.1	0.2	15.0	202.9
158.0	1482.9	- 804.7	0.3	15.0	204.5
160.0	1511.5	- 796.0	0.3	15.0	207.0
162.0	1539.4	- 784.8	0.4	15.0	211.4
164.0	1565.1	- 771.6	0.5	15.0	1000.0
166.0	1591.5	- 757.2	0.5	15.0	1000.0
168.0	1617.3	- 742.0	0.6	15.0	1000.0
180.0	1759.6	- 632.3	0.7	15.0	1000.0
192.0	1889.7	- 507.8	0.8	15.0	1000.0
204.0	2017.5	- 381.1	0.8	15.0	1000.0
216.0	2145.1	- 254.1	0.8	15.0	1000.0
228.0	2272.6	- 127.0	0.8	15.0	1000.0
240.0	2400.0	0.1	0.8	15.0	1000.0

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library U.S. Naval Postgraduate School Monterey, California 93940	2
3.	Mr. John Slaughter Navy Electronics Laboratory San Diego, California	1
4.	Mr. Lawrence Harvey Navy Electronics Laboratory San Diego, California	1
5.	Prof. Jack R. Borsting Code 55 U.S. Naval Postgraduate School Monterey, California 93940	15
6.	Prof. Frank D. Faulkner Code 53Fa U.S. Naval Postgraduate School Monterey, California 93940	75
7.	Mr. George D. Schmieg Apt. #1, 908 Rich Ave. Mountain View, California	10

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* U.S. Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

OPTIMUM SUBMARINE ROUTING II COMPUTATIONAL ROUTINES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Thesis, M.S., August 1966

5. AUTHOR(S) *(Last name, first name, initial)*

SCHMIEG, George D.

6. REPORT DATE August 1966	7a. TOTAL NO. OF PAGES 71	7b. NO. OF REFS 4

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

[redacted]

Mar 4/2/70

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

In this paper the details of computing an optimum route for a submarine are studied.

Typical functions representing the listening devices were used. It was found that in some cases several extremals existed and it was necessary to set up tests for the Legendre and Weierstrass conditions. The problem is further complicated by the fact that the optimum control-variables may lie on the boundary of the region of allowed values and further routines must be adjoined for this. Further, corners may occur and in particular the control may move discontinuously from a boundary point to an interior point or vice versa. The routines were made up to effect a compromise between the need for accuracy and reasonable computational time.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Optimum Submarine Routing Computational Routines						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8c, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.